



Hybrid Binary Dragonfly Algorithm with Simulated Annealing for Feature Selection

Hamouda Chantar¹ · Mohammad Tubishat² · Mansour Essgaer¹ · Seyedali Mirjalili^{3,4}

Received: 19 May 2020 / Accepted: 10 May 2021 / Published online: 25 May 2021
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2021

Abstract

There are various fields are affected by the growth of data dimensionality. The major problems which are resulted from high dimensionality of data including high memory requirements, high computational cost, and low machine learning classifier performance. Therefore, proper selection of relevant features from the set of available features and the removal of irrelevant features will solve these problems. Therefore, to solve the feature selection problem, an improved version of Dragonfly Algorithm (DA) is proposed by combining it with Simulated Annealing (SA), where the improved algorithm named BDA-SA. To solve the local optima problem of DA and enhance its ability in selecting the best subset of features for classification problems, Simulated Annealing (SA) was applied to the best solution found by Binary Dragonfly algorithm in attempt to improve its accuracy. A set of frequently used data sets from UCI repository was utilized to evaluate the performance of the proposed FS approach. Results show that the proposed hybrid approach, named BDA-SA, has superior performance when compared to wrapper-based FS methods including a feature selection method based on the basic version of Binary Dragonfly Algorithm.

Keywords Feature selection · Dragonfly algorithm · Simulated annealing algorithm · Optimization

Introduction

Recently, data mining field has become an active research area due to presence of the huge amount of data in digital format that needs to be transformed into useful information. The main task of data mining is to build models for the

discovery of useful hidden patterns in collections of huge data. It is considered as an essential step in the knowledge discovery process [16]. Preprocessing of data is a critical step in data mining. It has a direct impact on data mining techniques such as classification. It affects the quality of discovered patterns and the accuracy of the classification models [16, 25]. Feature selection is one of the main pre-processing steps in data mining that aims to discard noisy and irrelevant features while retaining the useful and informative ones. Selecting the ideal or near ideal subset of given features leads to accurate classification results and less computational cost [6, 25]. Feature selection approaches are classified based on estimation criteria of selected subset of features into two classes filter and wrapper approaches [6]. Wrapper techniques heavily rely on their search for optimal subset of features on the accuracy of machine learning classifiers such as KNN or SVM, whereas filter techniques use scoring matrices such as chi-square and information gain to assess the goodness of the selected subset of features. More precisely, in filter approaches, attributes are ranked using a filter approach, e.g., chi-square, and then the attributes with less than a predefined threshold are removed [1, 6, 14].

✉ Hamouda Chantar
hamoudak77@gmail.com

Mohammad Tubishat
mtubishat@yahoo.com

Mansour Essgaer
mansour.alssager@gmail.com

Seyedali Mirjalili
ali.mirjalili@gmail.com

- ¹ Faculty of Information Technology, Sebha University, Sebha, Libya
- ² School of Information Technology, Skyline University College, Sharjah, United Arab Emirates
- ³ Center for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia
- ⁴ Yonsei Frontier Lab, Yonsei University, Seoul, Korea

Generally speaking, finding an optimal subset of features is a challenging task. FS has gained the interest of many researchers in data mining and machine learning fields [8, 14]. The literature shows that meta-heuristic techniques have been very effective in tackling many optimization problems like machine learning, engineering design, data mining, production problems and feature selection [38]. For feature selection problem, Genetic algorithm (GA) and Particle swarm optimization algorithm (PSO) have been successfully utilized for solving many feature selection problems [6, 7, 13, 17]. Moreover, many biologically inspired approaches such as simulated annealing (SA) [29], Tabu search (TS) [45], Ant Colony Optimization (ACO) [22], Binary Bat Algorithm [32], Moth-Flame Optimization Algorithm [44], Antlion Optimization Algorithm [43], Dragonfly Optimization Algorithm [26], and Whale Optimization Algorithm (WOA) [35] have been efficiently applied to discover the best subsets of features for many classification problems.

As discussed in [37, 38], when designing and using meta-heuristic techniques, two main different criteria must be considered: diversification which refers to search space exploration, and intensification which means the exploitation of the optimal solution (e.g., best subset of features found so far). Based on these criteria, meta-heuristic techniques can be distinguished into two branches. Population-based meta-heuristics (e.g., PSO) and single-solution based techniques such as SA and TS which are biased towards exploitation. The performance of the search algorithm can be improved if an appropriate balance between the exploration and the exploitation is achieved. The desired balance between exploration and exploitation can be achieved via combining two techniques, e.g., population-based approach and single-solution based approach.

Various hybrid meta-heuristic approaches have been proposed for solving wide range of optimization problem. Hybrid approaches have also been popular, because they benefit from advantages of two or more algorithms [37]. In [18], to control the search procedure, local search approaches were embedded in GA algorithm. Obtained results show better performance in comparison with classical versions of GA. In [42], simulated annealing algorithm in conjunction with Genetic Algorithms were used to optimize industrial production management problem. In addition, a hybrid approach based on Markov chain and simulated annealing algorithms was designed to tackle the travel salesman problem [28]. Furthermore, in [4], Particle Swarm Optimization algorithm was hybridized with Simulated Annealing algorithm to avoid PSO in getting trapped in local optima. The designed approach was applied to deal with complex multi-dimensional optimization problem. Finally, in [3], the ACO algorithm was used in conjunction with a GA as a hybrid approach for feature selection in text classification domain. The proposed approach recorded better results in

comparison with filter approaches and the classic version of ACO. Despite the effectiveness that meta-heuristic algorithms have shown to solve many optimization problems, they have some shortcomings such as their sensitivity to parameter tuning, finding the optimal parameters for difference optimization problems is very important. Furthermore, in terms of performance, many of the present optimization algorithms have difficulties in dealing with high dimension optimization problems [5]. In addition, the computational cost of using meta-heuristics in general and the hybrid meta-heuristic approaches in particular is relatively high.

In this work, SA is used to enhance the best solution found so far (subset of features) by the Binary Dragonfly Optimizer. The classic version of Binary Dragonfly Algorithm was used for feature selection [26], but to best of our knowledge, this is the first time that Binary Dragonfly hybridized with Simulated Annealing is utilized in feature selection domain. The remaining of this paper is organized as follows: related work is reviewed in section “[Related Work](#)”, whereas section “[Algorithms](#)” presents the BDA and SA Algorithms. Section “[Proposed Feature Selection Method](#)” explains the proposed FS approach. In section “[Experiments](#)”, the experimental results are presented and discussed. Conclusion and future work are shown in section “[Conclusion](#)”.

In this paper, there are number of key contributions as follows:

- BDA-SA: an improved version of the standard binary version of BDA is proposed.
- The main improvement includes the combination of DA with SA to solve the local optima problem of standard BDA.
- BDA-SA wrapper feature selection model is developed in this paper.
- We evaluated and compared BDA-SA with a number of well-known algorithms including (BDA, PSO, BGWO and BALO).
- We compared the results with using all features, where we used 18 data sets from UCI repository which are frequently used by feature selection research. and ALO) using 18 benchmark data sets from UCI repository. From these results, it is clearly confirmed the superiority of BDA-SA in comparison to these baseline algorithms.

Related Work

Based on literature investigation, many optimization algorithms were improved by combining them with local search algorithms (LSA). For example, in a study by [11], Elgamal et al. improved Harris Hawks Optimization (HHO) Algorithm by SA and applied it for feature selection problem. In [2], the

authors also improved water cycle optimization with SA and applied it for spam email detection. In work by [41], performance of Salp Algorithm (SSA) was enhanced by combining it with a new developed LSA and applied it for feature selection problem. Also, in [40], WOA was combined with new a new local search algorithm to overcome the problem of local optima, and applied it for rules selection problem. Furthermore, in [21], Jia et al. improved spotted hyena optimization using SA and applied it for feature selection problem. Also, Simulated Annealing was hybridized with GA in [27] to be utilized as feature selection method for the classification of power disturbance in the Power Quality problem. In [33], Genetic Algorithm GA was used with SA to extract features to tackle the examination timetabling problem. A local search strategy was embedded in Particle Swarm Optimization algorithm to guide the PSO during the search process for the best subset of feature in classification [31]. Mafarja and Mirjalili [25] proposed two hybrid wrapper feature selection approaches based on Whale Optimization and Simulated Annealing algorithms. The aim of using SA is to enhance the exploitation of WOA. Results showed that the proposed approaches improved the classification accuracy in comparison with other wrapper feature selection techniques.

The literature also revealed that many successful efforts were done to improve the performance of Dragonfly optimization algorithm. For instance, in [19], Sayed et al. used several chaotic maps to adjust the movement parameters of dragonflies of DA algorithm through the iterations to accelerate its convergence rate. Hammouri et al. in [15] adopted several functions such as linear, quadratic, and sinusoidal for updating the main coefficients of Binary Dragonfly algorithm. Also, a hyper learning strategy was utilized by [39] to boost the Binary Dragonfly algorithm to avoid local optima, and enhance its search behaviour for an ideal subset of features. the proposed method was applied to a coronavirus disease (COVID-19) data set. Experimental results demonstrated the ability of hyper learning based BDA in improving the classification accuracy. Finally, in [34], Qasim et al. proposed a feature selection approach in which the binary dragonfly algorithm (BDA) was hybridized with statistical dependence (SD). The proposed hybrid approach confirmed its efficiency in increasing the classification accuracy. Thus, based on the achieved results and improvements conducted in these mentioned studies, it has motivated us to combine SA as a LSA with DA to improve its search ability for feature selection problem.

Algorithms

Dragonfly Algorithm

Dragonfly Algorithm is a recent biologically inspired optimization approach which was proposed by Seyedal

Mirjalili in 2015 [30]. It was found that dragonfly swarming behavior depends on two sorts of swarming behavior: hunting and migration [30, 36]. Hunting swarm of dragonflies moves in small subgroups over a restricted area to find and hunt preys. This behavior was utilized to simulate the exploration part of optimization process. In the migration behavior, in contrast with hunting swarm, dragonflies move a long one direction in bigger subgroups. This behavior was exploited to simulate the exploitation part of the optimization [30, 36]. Generally, the aim of swarm members is to co-operate to discover food places, and to protect themselves form danger of enemies. Based on these two aims, a set of factors is mathematically modeled for adjusting the positions of members in the swarm. The mathematical models for implementing the swarming behavior of dragonfly insects are given as follows [12, 26, 30]:

Separation indicates the way that flying dragonflies follow to avoid clashes between themselves. This can be mathematically written as in Eq. (1):

$$S_i = - \sum_{j=1}^N X - X_j \quad (1)$$

where X refers to the current search agent, while X_i denotes the j -th neighbor of X . N represents the number of neighbors. *Alignment* refers to the way of adjusting the velocity of an individual with respect to the velocity vector of other close dragonflies in the swarm. This can be mathematically written using Eq. (2):

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (2)$$

where V_j refers to the j -th neighbor's velocity vector.

Cohesion is a factor for position update of search agents that represents the desire of search agents to travel towards the mass center. It is mathematically written as in Eq. (3):

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (3)$$

Attraction denotes the interest of search agents to travel in direction of food location. The tendency of i -th member in the swarm to move towards the food source is obtained using Eq. (4):

$$F_i = F_{location} - X \quad (4)$$

where $F_{location}$ refers to the location of food source, and X refers to the current member.

Distraction refers to mechanism that dragonflies follow to flee from enemy. The distraction of i -th dragonfly is defined as in Eq. (5):

$$E_i = E_{location} + X \quad (5)$$

where $E_{location}$ presents current position of the enemy, and X is the position of the current member.

To find the optimum solution for a given optimization problem, DA defines a position vector and a step vector for each search agent in the swarm. These vectors are utilized to update the positions of search agents in the search space of the given optimization problem. The step vector which refers to travelling direction of dragonflies is formulated as follow [26, 30]:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (6)$$

where s , a , c , f , and e are known as weighting factors for separation (S_i), alignment (A_i), cohesion (C_i), attraction (F_i) and distraction (E_i) of the i -th search agent, respectively. w refers to the inertia weight.

The obtained step vector (ΔX) is used to estimate the position vector of search agent X as follows:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (7)$$

where t indicates the current iteration.

The basic version of Dragonfly optimizer is proposed for the problems in continuous search space. The dragonflies can update their position by adding the step vector to the position vector. Feature selection is a binary optimization problem, so the update strategy as in Eq. (7) is not possible for binary search space. Mirjalili [30] utilized the following transfer function to convert the step vector values to a number restricted in $[0,1]$.

$$T(\Delta x) = \left| \frac{\Delta x}{\sqrt{\Delta x^2 + 1}} \right| \quad (8)$$

The above transfer function is used to find the probability of updating the position of dragonflies in the swarm, and then the following equation is employed to update the positions of dragonflies (search agents):

$$X_{t+1} = \begin{cases} \neg X_t & r < T(\Delta x_{t+1}) \\ X_t & r \geq T(\Delta x_{t+1}) \end{cases} \quad (9)$$

where r is a number in the range of $[0,1]$.

Algorithm 1 presents the pseudocode of Binary Dragonfly algorithm.

Algorithm 1 Pseudocode of Binary Dragonfly Algorithm

```

Initialize positions of all search agents in the swarm  $X_i (i = 1, 2, \dots, n)$ 
Initialize step vectors of all search agents  $\Delta X_i (i = 1, 2, \dots, n)$ 
while (termination criteria is not reached) do
    Estimate all dragonflies' objective value.
    Update values of ( $F$ ) and ( $E$ )
    Update values of swarming factors
    (i.e.,  $w, s, a, c, f$ , and  $e$ )
    Use Eqs. from (1 to 5) to find  $S, A, C, F$ , and  $E$ 
    Update step vector of search agent ( $\Delta X_{t+1}$ ) using Eq.
    (6)
    Use Eq.(8) to find the probabilities.
    Update position vectors using Eq.(9)
end while
Return the optimum search agent

```

Simulated Annealing

SA is a single-solution based meta-heuristic optimization algorithm, introduced by Kirkpatrick et al. [23] in 1983. It has been widely used to tackle discrete and continuous optimization problems. SA is classified as a hill-climbing local search approach, in which a certain probability is used to decide weather to accept worse solution or not [23]. SA generates an initial solution (in our case, the best solution found so far by BDA is used as SA initial solution). A neighbour solution to the optimal one found so far is generated by SA based on fitness value of the neighbour and a specific neighbourhood structure. If the calculated fitness of neighbour is better (less or equal) than the fitness of optimal solution, then the neighbour solution is selected as the optimal one. Boltzmann probability, $P = e^{-\frac{\Phi}{T}}$ is applied as an acceptance condition of the neighbour solution. Φ refers to the difference between the fitness of the optimal and neighbour solutions, and T is named temperature which is gradually reduced based on cooling schedule throughout the search procedure [20, 23, 25]. In this paper, as adopted in [25], the initial temperature equals $2 * |N|$, where N is the number of features in each data set, and $T = 0.93 * T$, is applied to calculate the cooling schedule. Algorithm 2 presents the pseudocode of SA algorithm [25].

Algorithm 2 Pseudocode of Simulated Annealing Algorithm

```

Initialize temperature  $T_{initial} = 2 * |N|$ ,  $N$  is the number
of features in dataset.
 $Optimal_{Solution} \leftarrow Sol$  // Best solution found so far by
BDA.
 $\alpha(Optimal_{Solution}) \leftarrow \alpha(Sol)$  //  $\alpha$  refers to the quality of
the solution
while ( $T > T_{initial}$ ) do
  Generate randomly  $Trail_{Solution}$  in the neighbour of
   $Sol$ 
  Estimate  $\alpha(Trail_{Solution})$ 
  if  $\alpha(Trail_{Solution}) > \alpha(Optimal_{Solution})$ 
     $Sol \leftarrow Trail_{Solution}$ 
     $Optimal_{Solution} \leftarrow Trail_{Solution}$ 
     $\alpha(Sol) \leftarrow \alpha(Trail_{Solution})$ 
     $\alpha(Optimal_{Solution}) \leftarrow \alpha(Trail_{Solution})$ 
  else if ( $\alpha(Trail_{Solution}) = \alpha(Optimal_{Solution})$ )
    Estimate  $|Trail_{Solution}|$  and  $|Optimal_{Solution}|$ 
    if ( $|Trail_{Solution}| < |Optimal_{Solution}|$ )
       $Sol \leftarrow Trail_{Solution}$ 
       $Optimal_{Solution} \leftarrow Trail_{Solution}$ 
       $\alpha(Sol) \leftarrow \alpha(Trail_{Solution})$ 
       $\alpha(Optimal_{Solution}) \leftarrow \alpha(Trail_{Solution})$ 
    end if
  else // worse solution is accepted
    Estimate  $\phi = \alpha(Trail_{Solution}) - \alpha(Optimal_{Solution})$ 
    Generate a random number in range,  $P=[0,1]$ 
    if ( $P \leq e^{-\frac{\phi}{T}}$ )
       $Sol \leftarrow Trail_{Solution}$ 
       $\alpha(Sol) \leftarrow \alpha(Trail_{Solution})$ 
    end if
  end if
   $T = 0.93 * T$  // update  $T$ 
end while
Return  $Optimal_{Solution}$ 

```

Proposed Feature Selection Method

Feature selection problem is distinguished as a binary optimization problem, so binary vectors are used to represent the solutions. In this way, if the value of a specific cell in the binary vector is set to 1, then the corresponding feature is retained. Otherwise, that feature is ignored. The size of the binary vector is equal to the number of features in the data set. Dragonfly optimization algorithm is a newly introduced optimization approach. The basic version of binary DA algorithm was used for feature selection in [26]. The main aim of this work is to improve the performance of BDA for feature solution problem. To achieve that purpose, the best solution obtained so far by BDA is passed to the SA algorithm to be used as an initial solution instead of the random generation of the initial solution. Therefore, SA will conduct a local search starting with the optimal solution found so far by BDA in attempt to find a better one. Figure 1 presents the flowchart of the proposed approach.

Feature selection is a multi-objective optimization problem, where the maximum accuracy of the classifier and least

number of selected features are two related objectives need to be achieved. Eq. (10) is commonly used as fitness function for feature selection [6, 25, 26].

$$Fitness = (\alpha * er) + (\beta * (\frac{m}{N})) \quad (10)$$

where er denotes the error rate of KNN machine learning classifier using the selected subset of features. α and β are two parameters used to make a balance between the classification accuracy and the size of subset of features (selected by the search agent), α is a number restricted in the range [0,1], and β equals $1 - \alpha$. N refers to the total number of features in the data set, and m indicates the cardinality of the subset of features selected by the search agent. In this work, since we are mostly interested in getting the highest classification accuracy, α is set equal to 0.99 as in the previous work [26]. The value α is set 0.99, because in this work we improved the binary version of DA algorithm which was developed in [26], and we used the same setting in our experiments to compare our results with the results were reported in BDA [26].

In general, diversification has larger importance than intensification in exploring potentially useful areas of the feature space especially at the beginning of the search process. In later phases, exploitation has larger importance, because the search for better solutions around the best one found by the exploration phase is required [24]. Hybrid approaches such as BDA-SA in our case can be used to achieve the desired balance between exploring and exploiting the search space. However, in comparison with classic wrapper approaches, where a heuristic technique and an evaluator are used, the computational cost of utilizing hybrid approaches is higher.

Experiments

Data Sets

In this work, 18 data sets from UCI repository were used to assess the performance of the proposed feature selection approach [10]. They are the same data sets used by many researchers to evaluate various feature selection approaches. Table 1 outlines the details of applied data sets for evaluating the proposed Binary Dragonfly algorithm based feature selection approach.

Parameter Settings

As in [26], each data set is split into three equal sets: training set, validation set, and test set. In addition, K-fold-cross-validation procedure is used to evaluate KNN classifier (the parameter K of KNN classifier is set to five as adopted in

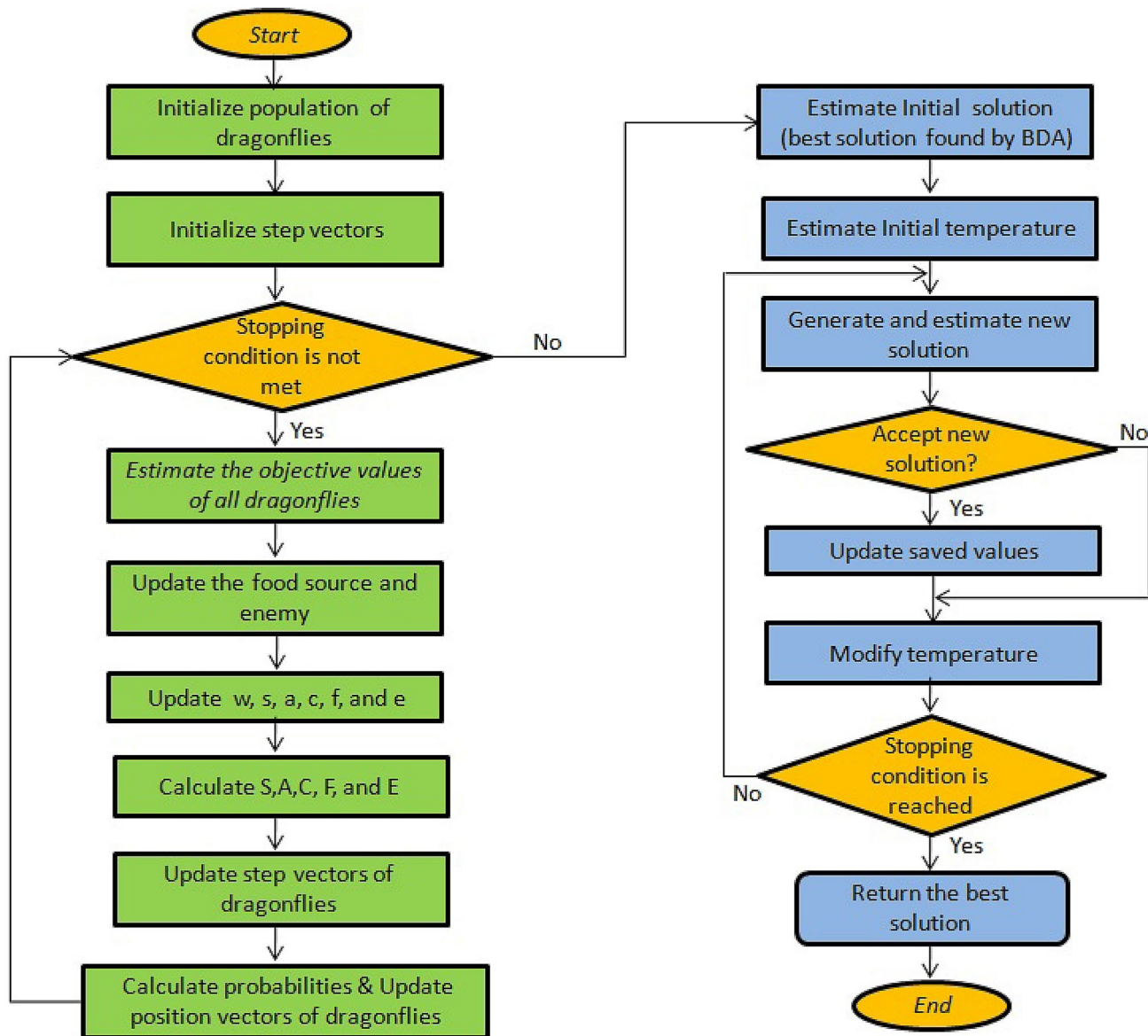


Fig. 1 Flowchart of BDA-SA algorithm

[26]). Also, results of several metaheuristics-based wrapper feature selection algorithms including Binary Particle Swarm Optimization (BPSO), Binary Ant Lion Optimization (BALO), and Binary Gray Wolf Optimization (BGWO) were also used for comparison purpose. Furthermore, for Binary Dragonfly algorithm, the original paper of Dragonfly algorithm [30] comprehensively studied the appropriate values of swarming factors and the inertia weight, for that reason, the same best parameter settings reported in that paper were adopted in this work. Moreover, the same values of parameters adopted in [25] for SA algorithm were used. The parameters of BGWO, BPSO, and BALO algorithms were selected based on recommended setting reported in the original publications and related studies in feature

selection domain. In all conducted experiments, common parameters were set as in Table 2. These values were set following a range of initial experiments. Comparison between approaches were made based on three criteria comprising classification accuracy, number of selected features, and best fitness. In addition, each approach was run 20 times with random initial solutions on a machine with Intel Core i5 processor 2.2 GHz and 4 GB of RAM.

Results and Discussion

This section presents all recorded results obtained from the proposed FS approach. The proposed hybrid approach BDA-SA was compared to the original version of BDA

Table 1 Data sets used in the experiments

Data set	Features	Instances
Breastcancer	9	699
BreastEW	30	569
CongressEW	16	435
Exactly	13	1000
Exactly2	13	1000
HeartEW	13	270
IonosphereEW	34	351
KrvskpEW	36	3196
Lymphography	18	148
M-of-n	13	1000
PenglungEW	325	73
SonarEW	60	208
SpectEW	22	267
Tic-tac-toe	9	958
Vote	16	300
WaveformEW	40	5000
WineEW	13	178
Zoo	16	101

Table 2 Parameter setting of algorithms

Parameter	Value
K parameter for KNN classifier	5
α parameter for fitness function	0.99
Number of search agents	10
Maximum number of iterations	100
a of BGWO	from 2 to 0
ω of BPSO	from 0.9 to 0.2
$C1$ and $C2$ parameters for velocity of BPSO	2.0

based approach. In terms of classification accuracy, it is clear from Table 3 that BDA-SA is able to classify most accurately on all data sets. It can be stated that SA succeeded to enhance the best solution found by BDA algorithm. In addition, in terms of best fitness, Table 3 reveals the averages of best fitness for the applied FS approaches on each data set. In most of the cases, BDA-SA obtained the lowest average of fitness value. BDA is slightly better than BDA-SA in only two cases (SonerEW and M-of-n datasets). Although, we will see later that the differences are not statistically significant. It is clear from Table 3 that BDA-SA has less averages of selected features than BDA in some cases, while the averages recorded by BDA are less on other cases. We previously observed that BDA-SA is superior on all cases in terms of classification accuracy. It is evident that when the average of selected features by BDA-SA is greater than BDA, this means that BDA-SA approach managed to find informative and relevant features

ignored by BDA, and when the average of selected features by BDA-SA is less than BDA, that means BDA-SA approach may removed some noisy or irrelevant features selected by BDA approach.

Since the main aim of this work is to enhance the performance of BDA algorithm by hybridizing it with SA algorithm, we conducted further statistical analysis to demonstrate that the hybrid BDA-SA approach is better than using the basic BDA alone for feature selection problem. Table 4 presents the standard deviation values of the averages of classification accuracy, best fitness, and number of selected features for BDA and BDA-SA approaches on each data set. In terms of classification accuracy, as in Table 4, it can be observed that BDA-SA approach behaves more robust than BDA based approach on almost all data sets. In terms of best fitness and number of selected features, as shown in Table 4, it can be seen that BDA-SA approach is better than BDA approach on half of the cases.

The average and standard deviation were used as measures to compare the overall results obtained from BDA and BDA-SA. To see whether the differences in the results are statistically significant or not, the non-parametric Wilcoxon test with significant level 0.05 was applied. This test is appropriate to compare the algorithms that have stochastic behaviour [9]. As in Table 5, the p values of the accuracy and the fitness show that BDA-SA recorded significantly better results than BDA on most of the data sets. In terms of selected features, the differences are statistically significant on eight cases, while on other data sets including BreastEW, CongressEW, WaveformEW, SpectEW, and IonosphereEW, p values show that the differences are not statistically significant. The superiority of BDA-SA particularly in terms of classification accuracy is expected, since it utilizes two powerful searching algorithms, DA which is efficient in exploration and SA that has a strong exploitation capability. The ability of DA is utilized in exploring the highly relevant regions in the feature space and avoid the trap of local optima, and then SA is used to intensify the nearby regions to the optimal solution (best subset of features) discovered by BDA algorithm. However, in terms of computational time, as revealed in Fig. 2, in all cases, the computation cost of BDA-SA is higher compared to BDA.

The performance of BDA-SA was also compared with three meta-heuristic-based feature selection approaches including Binary Particle Swarm Optimization, Binary Ant Lion Optimization, and Binary Gray Wolf Optimization. In terms of accuracy rates, as revealed in Table 6, BDA-SA outperformed all its competitors. In addition, in terms of best fitness rates, as presented in Table 7, BDA-SA recorded lowest averages of best fitness on fifteen out of eighteen data sets. Furthermore, in terms of lowest number of selected features, Table 8 shows that

Table 3 Averages of classification accuracy, best fitness and selected features obtained from BDA and BDA-SA

Data set	Accuracy		Best Fitness		Selected Features	
	BDA	BDA-SA	BDA	BDA-SA	BDA	BDA-SA
Breastcancer	0.968	0.988	0.038	0.018	6.000	6.000
BreastEW	0.960	0.974	0.043	0.030	13.500	12.450
CongressEW	0.967	0.975	0.035	0.028	5.000	4.150
Exactly	0.982	1.000	0.023	0.005	6.250	6.000
Exactly2	0.744	0.759	0.244	0.240	1.450	1.450
HeartEW	0.842	0.895	0.159	0.109	5.950	6.900
IonosphereEW	0.919	0.929	0.079	0.074	11.650	10.400
KrvskpEW	0.958	0.976	0.031	0.028	18.350	14.400
Lymphography	0.872	0.911	0.131	0.092	7.400	8.500
M-of-n	0.995	0.997	0.006	0.008	6.200	6.150
PenglungEW	0.909	0.930	0.093	0.049	123.350	141.300
SonarEW	0.914	0.917	0.077	0.086	27.550	23.900
SpectEW	0.857	0.866	0.143	0.137	8.650	8.850
Tic-tac-toe	0.784	0.818	0.207	0.189	8.200	7.800
BDA Vote	0.953	0.964	0.049	0.037	6.550	3.050
WaveformEW	0.755	0.805	0.236	0.198	21.000	21.100
WineEW	0.987	0.999	0.009	0.008	6.200	8.850
Zoo	0.959	0.979	0.042	0.024	8.350	5.650

Bold values represent the best results

Table 4 Standard deviation of the averages of classification accuracy, best fitness and selected features obtained from BDA and BDA-SA

Data set	Accuracy		Best Fitness		Selected Features	
	BDA	BDA-SA	BDA	BDA-SA	BDA	BDA-SA
Breastcancer	0.001278	0.000639	0.000000	0.000633	0.000000	0.000000
BreastEW	0.005140	0.004020	0.004360	0.004160	2.013110	2.665000
CongressEW	0.004183	0.003490	0.003603	0.003320	1.716790	1.039900
Exactly	0.063000	0.000000	0.063070	0.000000	0.910460	0.000000
Exactly2	0.035800	0.001573	0.000333	0.000498	1.234300	1.394500
HeartEW	0.014274	0.003040	0.008007	0.003183	0.998683	0.640720
IonosphereEW	0.012887	0.010824	0.009205	0.011020	2.680800	2.186000
KrvskpEW	0.041039	0.003435	0.005190	0.003648	2.518876	2.542270
Lymphography	0.011371	0.008173	0.011450	0.008092	1.142481	1.960129
M-of-n	0.016938	0.008141	0.003192	0.008298	0.410391	0.366348
PenglungEW	0.015869	0.028996	0.015586	0.016101	7.727429	8.676041
SonarEW	0.021943	0.018561	0.011350	0.018594	4.006245	3.567396
SpectEW	0.008352	0.008255	0.005123	0.008304	1.225819	2.433862
Tic-tac-toe	0.038510	0.003962	0.010051	0.001797	1.641565	1.880649
Vote	0.006840	0.003262	0.002680	0.003612	2.459675	1.234376
WaveformEW	0.028255	0.006023	0.000938	0.006278	2.991215	2.174009
WineEW	0.030413	0.003458	0.004565	0.002595	1.576138	1.814416
Zoo	0.016711	0.004384	0.013161	0.004260	1.565248	1.565200

Bold values represent the best results

BDA-SA outperformed other algorithms on more than 50% of tested cases. Also, the average of computational

time for each approach was considered. Figure 3 shows the computational cost of BDA-SA, BPSO, BGWO, and

Table 5 P values of the Wilcoxon ranksum test over 20 runs for classification accuracy, Best Fitness and Selected Features of BDA and BDA-SA ($P \geq 0.05$ have been underlined)

Data set	Accuracy	Best fitness	Selected features
Breastcancer	0.000080	0.000080	N/A
BreastEW	0.000080	0.000080	<u>0.170680</u>
CongressEW	0.000540	0.000320	<u>0.107400</u>
Exactly	N/A	N/A	N/A
Exactly2	0.000140	0.000080	N/A
HeartEW	0.000080	0.000080	0.006140
IonosphereEW	0.024440	<u>0.087260</u>	<u>0.123560</u>
KrvskpEW	<u>0.073460</u>	<u>0.067240</u>	0.000500
Lymphography	0.00008	0.00008	0.047700
M-of-n	N/A	N/A	N/A
PenglungEW	0.013900	0.000100	0.000160
SonarEW	<u>0.624140</u>	<u>0.204080</u>	0.020880
SpectEW	0.023200	0.044440	<u>0.968100</u>
Tic-tac-toe	0.000080	0.000080	N/A
Vote	0.000200	0.000080	0.000420
WaveformEW	0.000080	0.000080	<u>0.703940</u>
WineEW	<u>0.058760</u>	<u>0.779480</u>	0.001000
Zoo	0.000640	0.000080	0.001420

The underline values mean that there is no significant difference

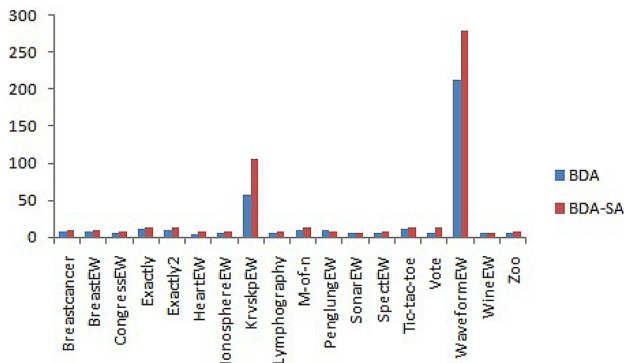


Fig. 2 Averages of computational time for BDA and BDA-SA

BALO. It can be observed that BPSO is the best in terms of lowest computational time.

Conclusion

This work introduced BDA-SA as a hybrid feature selection approach. The main goal was to enhance the performance of Binary Dragonfly algorithm especially in terms for classification accuracy. The best solution found so far by BDA algorithm was used as initial solution by SA algorithm to conduct a local search to find better solution than the one obtained by BDA. The proposed approach was

Table 6 Comparison between BDA-SA and other algorithms in terms of classification accuracy

Data set	BPSO	BALO	BGWO	BDA-SA
Breastcancer	0.949	0.931	0.957	0.988
BreastEW	0.924	0.930	0.935	0.974
CongressEW	0.912	0.915	0.923	0.975
Exactly	0.672	0.626	0.666	1.000
Exactly2	0.725	0.702	0.717	0.759
HeartEW	0.789	0.751	0.751	0.895
IonosphereEW	0.845	0.813	0.787	0.929
KrvskpEW	0.850	0.761	0.857	0.976
Lymphography	0.730	0.684	0.786	0.911
M-of-n	0.814	0.733	0.800	0.997
PenglungEW	0.728	0.789	0.724	0.930
SonarEW	0.799	0.778	0.796	0.917
SpectEW	0.808	0.831	0.787	0.866
Tic-tac-toe	0.712	0.694	0.707	0.818
Vote	0.905	0.882	0.915	0.964
WaveformEW	0.743	0.701	0.743	0.805
WineEW	0.916	0.916	0.895	0.999
Zoo	0.846	0.808	0.829	0.979

Bold values represent the best results

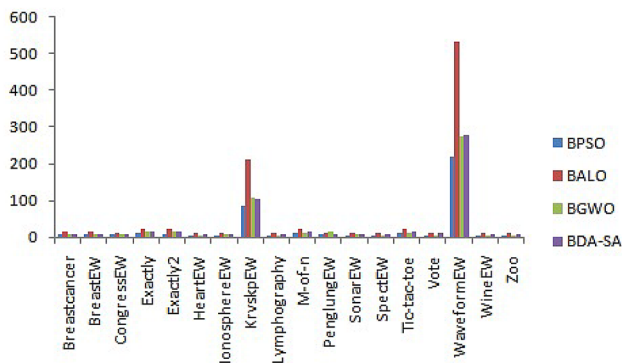
Table 7 Comparison between BDA-SA and other algorithms in terms of best fitness

Data set	BPSO	BALO	BGWO	BDA-SA
Breastcancer	0.038	0.038	0.031	0.018
BreastEW	0.044	0.036	0.045	0.030
CongressEW	0.034	0.045	0.030	0.028
Exactly	0.037	0.164	0.163	0.005
Exactly2	0.243	0.257	0.246	0.240
HeartEW	0.135	0.165	0.166	0.109
IonosphereEW	0.113	0.141	0.176	0.074
KrvskpEW	0.030	0.045	0.049	0.028
Lymphography	0.145	0.156	0.105	0.092
M-of-n	0.005	0.046	0.042	0.008
PenglungEW	0.165	0.122	0.216	0.049
SonarEW	0.093	0.123	0.126	0.086
SpectEW	0.134	0.096	0.127	0.137
Tic-tac-toe	0.201	0.217	0.191	0.189
Vote	0.032	0.062	0.046	0.037
WaveformEW	0.200	0.219	0.217	0.198
WineEW	0.018	0.008	0.038	0.008
Zoo	0.045	0.033	0.084	0.024

assessed on a set of frequently used data sets from UCI machine learning repository. The performance of BDA-SA was compared to the native BDA algorithm as well as various algorithms comprising BGWO, BPSO, and BALO. Experimental results show that BDA-SA outperformed

Table 8 Comparison between BDA-SA and other algorithms in terms of selected features

Data set	BPSO	BALO	BGWO	BDA-SA
Breastcancer	6.000	6.000	4.000	6.000
BreastEW	12.800	22.600	20.600	12.450
CongressEW	3.200	9.300	9.500	4.150
Exactly	6.200	7.700	7.750	6.000
Exactly2	2.050	8.450	8.200	1.450
HeartEW	3.550	8.900	8.500	6.900
IonosphereEW	12.300	25.500	20.550	10.400
KrvskpEW	16.850	26.900	24.300	14.400
Lymphography	6.050	11.050	11.800	8.500
M-of-n	6.000	7.500	7.650	6.150
PenglungEW	146.250	253.650	204.700	141.300
SonarEW	27.000	47.450	40.050	23.900
SpectEW	8.000	15.350	13.750	8.850
Tic-tac-toe	8.900	9.000	6.000	7.800
Vote	5.000	8.500	8.100	3.050
WaveformEW	21.150	32.400	30.600	21.100
WineEW	6.450	10.250	8.400	8.850
Zoo	8.150	10.550	9.350	5.650

**Fig. 3** Comparison between BDA-SA and other algorithms in terms of computational time

BDA and the other algorithms. In the future, it is worth to evaluate the proposed hybrid approach on more complex data sets.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s42979-021-00687-5>.

Declarations

Conflicts of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Ahmed S, Mafarja M, Faris H, Aljarah I. Feature selection using salp swarm algorithm with chaos. In: Proceedings of the 2Nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, ISMSI '18, pp. 65–69. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3206185.3206198>.
- Al-Rawashdeh G, Mamat R, Hafhizah BARN. Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection. IEEE Access. 2019;7:143721–34. <https://doi.org/10.1109/ACCESS.2019.2944089>.
- Basiri ME, Nemati S. A novel hybrid aco-ga algorithm for text feature selection. In: 2009 IEEE Congress on Evolutionary Computation, pp. 2561–2568. IEEE (2009)
- Basu M, Deb P, Garai G. Hybrid of particle swarm optimization and simulated annealing for multidimensional function optimization. Int J Inform Technol. 2014;20(1).
- Boussaïd I, Lepagnot J, Siarry P. A survey on optimization metaheuristics. Inf Sci. 2013;237:82–117.
- Chantar HK, Corne DW Feature subset selection for arabic document categorization using BPSO-KNN. In: Nature and Biologically Inspired Computing, 2011 Third World Congress on, pp. 546–551. IEEE (2011)
- Chuang LY, Yang CH, Li JC. Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput. 2011;11:239–48. <https://doi.org/10.1016/j.asoc.2009.11.014>.
- Dash M, Liu H. Feature selection for classification. Intell Data Anal. 1997;1:131–56.
- Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput. 2011;1:3–18.
- Dua D, Graff C. UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
- Elgamal M, Binti N, Tubishat M, Alswaiti M, Mirjalili S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. IEEE Access. 2020;8. <https://doi.org/10.1109/ACCESS.2020.3029728>.
- Elhariri E, El-Bendary N, Hassanien A.E. Bio-inspired optimization for feature set dimensionality reduction. In: 2016 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pp. 184–189 (2016)
- Ferri F.J, Kadirkamanathan V, Kittler J. Feature subset search using genetic algorithms. In: IEE/IEEE Workshop on Natural Algorithms in Signal Processing, IEE. Press (1993)
- Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003; 3,1157–1182. <http://dl.acm.org/citation.cfm?id=944919.944968>
- Hammouri A, Mafarja M, Al-Betar M, Awadallah M, Doush I. An improved dragonfly algorithm for feature selection. Knowl-Based Syst. 2020;203:106131. <https://doi.org/10.1016/j.knsys.2020.106131>.
- Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Amsterdam: Elsevier; 2012.

17. Huang CL, Dun JF. A distributed pso-svm hybrid system with feature selection and parameter optimization. *Appl Soft Comput.* 2008;8:1381–91. <https://doi.org/10.1016/j.asoc.2007.10.007>.
18. Il-Seok Oh, Jin-Seon Lee, Byung-Ro Moon: Hybrid genetic algorithms for feature selection. *IEEE Trans Pattern Anal Mach Intell.* 2004;26(11):1424–37.
19. Ismail Sayed G, Tharwat A, Hassanien AE. Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. *Appl Intell.* 2019;49. <https://doi.org/10.1007/s10489-018-1261-8>.
20. Jeong IS, Kim HK, Kim TH, Hwi LD, Kim K, Kang SH. A feature selection approach based on simulated annealing for detecting various denial of service attacks. *Softw Netw* 2016; 173–190 (2016)
21. Jia H, Li J, Song W, Peng X, Lang C, Li Y. Spotted hyena optimization algorithm with simulated annealing for feature selection. *IEEE Access* PP, 1–1 (2019). <https://doi.org/10.1109/ACCESS.2019.2919991>
22. Kashef S, Nezamabadi-pour H. An advanced aco algorithm for feature subset selection. *Neurocomputing.* 2015;147:271–9. <https://doi.org/10.1016/j.neucom.2014.06.067>.
23. Kirkpatrick S, D Jr Gelatt C, P. Jr. Vecchi M. Optimization by simulated annealing. *Science (New York, N.Y.)* 1983;220,671–80. <https://doi.org/10.1126/science.220.4598.671>
24. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier Viger P, Li X, Mirjalili S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl-Based Syst.* 2018;161:185–204.
25. Mafarja M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing.* 2017;. <https://doi.org/10.1016/j.neucom.2017.04.053>.
26. Mafarja M.M, Eleyan D, Jaber I, Hammouri A, Mirjalili S. Binary dragonfly algorithm for feature selection. In: 2017 International Conference on New Trends in Computing Sciences (ICTCS), pp. 12–17 (2017)
27. Manimala K, Selvi K, Ahila R. Hybrid soft computing techniques for feature selection and parameter optimization in power quality data mining. *Appl. Soft Comput.* 2011;11:5485–97.
28. Martin OC, Otto SW. Combining simulated annealing with local search heuristics. *Ann OR.* 1993;63:57–75.
29. Meiri R, Zahavi J. Using simulated annealing to optimize feature selection problem in marketing applications. *Eur J Oper Res.* 2006;171:842–58. <https://doi.org/10.1016/j.ejor.2004.09.010>.
30. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl.* 2015;27(4):1053–73.
31. Moradi P, Gholampour M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Appl Soft Comput.* 43(C), 117–130 (2016). <https://doi.org/10.1016/j.asoc.2016.01.044>.
32. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang XB. A binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, 2012; pp. 291–297
33. Olabiyisi Stephen O, Fagbola Temitayo M, Omidiora Elijah O, Oyeleye Akin C. Hybrid metaheuristic feature extraction technique for solving timetabling problem. *Int J Sci Eng Res USA.* 2012;3(8):1–6.
34. Qasim O, Sabah M, Alzamzum F. Hybrid binary dragonfly optimization algorithm with statistical dependence for feature selection. *Int J Math Eng Manag Sci.* 2020;5:1420–8. <https://doi.org/10.33889/IJMEMS.2020.5.6.105>.
35. Sharawi M, Zawbaa H.M, Emary E, Zawbaa H.M, Emary E. Feature selection approach based on whale optimization algorithm. In: 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), pp. 163–168 (2017)
36. Song J, Li S. Elite opposition learning and exponential function steps-based dragonfly algorithm for global optimization. In: 2017 IEEE International Conference on Information and Automation (ICIA), pp. 1178–1183 (2017)
37. Talbi EG. A taxonomy of hybrid metaheuristics. *J Heuristics.* 2002;8:541–64. <https://doi.org/10.1023/A:1016540724870>.
38. Talbi EG Metaheuristics: from design to implementation. Wiley (2009). <https://hal.inria.fr/hal-00750681>
39. Too J, Mirjalili S. A hyper learning binary dragonfly algorithm for feature selection: A covid-19 case study. *Knowl-Based Syst.* 2020;212:106553. <https://doi.org/10.1016/j.knosys.2020.106553>.
40. Tubishat M, Idris N, Abushariah M. Explicit aspects extraction in sentiment analysis using optimal rules combination. *Future Gen Compu Syst.* 2020;114. <https://doi.org/10.1016/j.future.2020.08.019>.
41. Tubishat M, Jaafar S, Alswaiti M, Mirjalili S, Idris N, Ismail MA, Omar M. Dynamic salp swarm algorithm for feature selection. *Expert Syst Appl.* 2020;164:113873. <https://doi.org/10.1016/j.eswa.2020.113873>.
42. Vasant P. Hybrid simulated annealing and genetic algorithms for industrial production management problems. *Int J Comput Methods (IJCM)* (2), 7 (2010)
43. Zawbaa H.M, Emary E, Parv B. Feature selection based on antlion optimization algorithm. In: 2015 Third World Conference on Complex Systems (WCCS), pp. 1–7 (2015)
44. Zawbaa H.M, Emary E, Parv B, Sharawi M. Feature selection approach based on moth-flame optimization algorithm. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4612–4617 (2016)
45. Zhang H, Sun G. Feature selection using tabu search method. *Pattern Recognit.* 2002;35:701–11. [https://doi.org/10.1016/S0031-3203\(01\)00046-2](https://doi.org/10.1016/S0031-3203(01)00046-2).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.