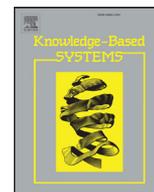




Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems

Majdi Mafarja^a, Ibrahim Aljarah^{b,*}, Ali Asghar Heidari^c, Abdelaziz I. Hammouri^d, Hossam Faris^b, Ala' M. Al-Zoubi^b, Seyedali Mirjalili^e

^a Department of Computer Science, Birzeit University, Birzeit, Palestine

^b King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

^c School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran

^d Department of Computer Information Systems, Al-Balqa Applied University, Al-Salt, Jordan

^e School of Information and Communication Technology, Griffith University, Nathan, Brisbane QLD 4111, Australia

ARTICLE INFO

Article history:

Received 25 September 2017

Revised 4 December 2017

Accepted 30 December 2017

Available online xxx

Keywords:

Grasshopper Optimization Algorithm
GOA

Feature selection

Classification

Metaheuristics

Evolutionary Population Dynamics

Optimization

ABSTRACT

Searching for the optimal subset of features is known as a challenging problem in feature selection process. To deal with the difficulties involved in this problem, a robust and reliable optimization algorithm is required. In this paper, Grasshopper Optimization Algorithm (GOA) is employed as a search strategy to design a wrapper-based feature selection method. The GOA is a recent population-based metaheuristic that mimics the swarming behaviors of grasshoppers. In this work, an efficient optimizer based on the simultaneous use of the GOA, selection operators, and Evolutionary Population Dynamics (EPD) is proposed in the form of four different strategies to mitigate the immature convergence and stagnation drawbacks of the conventional GOA. In the first two approaches, one of the top three agents and a randomly generated one are selected to reposition a solution from the worst half of the population. In the third and fourth approaches, to give a chance to the low fitness solutions in reforming the population, Roulette Wheel Selection (RWS) and Tournament Selection (TS) are utilized to select the guiding agent from the first half. The proposed GOA-EPD approaches are employed to tackle various feature selection tasks. The proposed approaches are benchmarked on 22 UCI datasets. The comprehensive results and various comparisons reveal that the EPD has a remarkable impact on the efficacy of the GOA and using the selection mechanism enhanced the capability of the proposed approach to outperform other optimizers and find the best solutions with improved convergence trends. Furthermore, the comparative experiments demonstrate the superiority of the proposed approaches when compared to other similar methods in the literature.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

The existence of thousands of applications of information systems complicated the role of extracting useful information from the collected data [1,2]. Data mining plays the main role in extracting the useful knowledge from the collected datasets [3,4]. The collected datasets may contain irrelevant and redundant data. Feature selection (FS) is one of the major preprocessing phases that aims to exclude the irrelevant/redundant data from the dataset being processed [5,6].

FS methods can be broadly categorized into three main classes: supervised [7], unsupervised [8], and semi-supervised methods [9]. Supervised FS requires the availability of the class labels to select proper features and used for classification problems. While in unsupervised FS, the class labels are not required, and used for clustering tasks. On the other hand, semi-supervised methods applied when part of the data is labeled.

There are several supervised, semi supervised, and unsupervised FS algorithms in literature. To name a few, the correlation-based feature selection (CFS) [7], fast correlation-based filter (FCBF) [10], and wavelet power spectrum (Spectrum) [11] are examples on supervised techniques. While non-negative spectral learning and sparse regression-based dual-graph regularized (NSSRD) feature selection is one of the latest unsupervised techniques proposed by Shang et al. in 2017 [8]. The subspace learning-based graph regularized (SGFS) technique and self-representation based

* Corresponding author.

E-mail addresses: mmafarja@birzeit.edu (M. Mafarja), i.aljarah@ju.edu.jo (I. Aljarah), as_heidari@ut.ac.ir (A.A. Heidari), Aziz@bau.edu.jo (A.I. Hammouri), hossam.faris@ju.edu.jo (H. Faris), a.alzoubi@jisdf.org (Ala' M. Al-Zoubi), seyedali.mirjalili@griffithuni.edu.au (S. Mirjalili).

<https://doi.org/10.1016/j.knosys.2017.12.037>

0950-7051/© 2017 Elsevier B.V. All rights reserved.

dual-graph regularized feature selection clustering (DFSC) are also well-established FS techniques proposed by Shang et al. in 2016 [12,13]. On the other hand, feature selection via spectral analysis, and forward feature selection [9,14] are examples on semi-supervised FS.

FS process can be accomplished in four major steps [15]: subset generation, subset assessment, ending criterion, and validation. From the evaluation perspectives, FS methods can be divided to two groups based on selection strategy: wrapper-based and filter-based. In filter-based methods, the selection of a subset is performed independently from the learning algorithm (e.g., classification). The merits of a feature or a subset of them is estimated with regard to specific characteristics of the information [16]. Examples of filter models include Chi-Square [17], Information Gain (IG) [18], Gain Ratio [19], and ReliefF [20]. In the wrapper-based methods, the goodness of a subset is evaluated based on a learning algorithm [21]. Examples of wrapper models include the LVW algorithm [22] and a neural network-based method [23].

Subset generation is considered as a search process to select a subset of items from the initial set using complete, heuristic search, or a random search [15,24,25]. The complete search generates all possible subsets to select the best one. If the dataset includes n features, then 2^n subsets will be generated and assessed, which is computationally expensive for the larger size datasets. Random search is another possible policy to select the attributes. It searches for the next feature subset randomly [26]. The main drawback of the random search strategy is that it may perform as a complete search in the worst case [5,27].

An alternative strategy to the previous two strategies is the heuristic search. Heuristic search can be clarified as a 'depth first' search managed by heuristics. According to Talbi [27], metaheuristic search methods can be defined as "upper level general methodologies (templates) that can be used as guiding strategies in designing underlying heuristics to solve specific optimization problems" [27]. Various metaheuristics such as Grey wolf optimizer (GWO) [28,29], Whale Optimization Algorithm (WOA) [30], Ant Lion Optimization (ALO) [31], Firefly Algorithm (FA) [32], Particle Swarm Optimization (PSO) [33], and Ant Colony Optimization (ACO) [34] may demonstrate superior efficiencies in tackling feature selection problems when compared to the exact methods [35,36]. Metaheuristic algorithms have shown improved results and efficiencies in dealing with many real-life applications such as path planning [37], clustering [38–40], neural network optimization [41–44] and power dispatch [45]. For example, E.S. Ali et al. applied the ALO to find the best location and sizing of renewable distributed generations [46]. Wu et al. utilized the WOA for path planning of solar-powered UAV [37]. Faris et al. also reviewed the recent variants and applications of the GWO [47]. The history of metaheuristics is presented in [48].

The GOA is a new efficient nature-inspired population-based metaheuristic algorithm [49] proposed by Saremi et al. in 2017 to inspire the idealized swarming behaviors of grasshopper insects in nature. This algorithm can disclose improved results and efficiencies on global unconstrained/constrained optimization and various real-life tasks. The basic GOA has been applied to realize the best parameters of proton exchange membrane fuel cells (PEMFCs) stack and the results exposed the viability of the GOA-based algorithm in dealing with the steady-state and dynamic models [50]. In 2017, Wu et al. [51] proposed a dynamic GOA for optimizing the distributed trajectory of UAVs in urban environments. They proved that this algorithm can attain enhanced results and satisfactory trajectories. Tharwat et al. [52] developed a modified multi-objective GOA (MOGOA) with external archive for constrained and unconstrained problems. Mirjalili et al. [53] also developed the basic multi-objective GOA and revealed that the proposed algorithm can tackle several benchmark problem, effectively and with better

performance in terms of accuracy of Pareto optimal solutions and the related distribution.

Although the metaheuristic algorithms do not guarantee finding the best solution in all runs, they can find relatively accurate solutions in a reasonable time [27,54]. Metaheuristics can be classified into two main families; single-solution and population-based algorithms [27,55]. In the former class (e.g., Simulated Annealing), one solution is manipulated and transformed during the search process, while a set of solutions is evolved in the former class (e.g., PSO). Single-solution-based algorithms show more exploitative behaviour; which means digging the space around a possible solution whereas the population-based class are more explorative or a mix of both behavior; which means exploring different regions of the space [27]. When designing a metaheuristic algorithm, these two criteria should be taken into account. High exploration decreases the quality of results and causes an unpromising convergence. This results in a failure to find the target global optimum. However, high exploitation may cause the optimizer to be trapped in Local Optima (LO).

Evolutionary algorithms (EA) are deep-rooted metaheuristics inspired by natural processes [56,57]. Genetic algorithms (GA), by J. H. Holland [58]; and evolutionary programming by L. Fogel et al. [59] are two different kind of EA. In recent years, many EA are proposed to tackle the optimization problems especially in the field of feature selection [60–62]. Ant Colony (AntRSAR) and Genetic Algorithm (GenRSAR) are two EAs that have been proposed by Jensen and Shen [63,64] and applied to FS problems. For instance, a chaos-based genetic FS method (CGFSO) has been proposed in [65]. Two hybrid approaches have been proposed in [66] between the GA and Simulated Annealing (SA) and in [67] between the GA and Record to Record algorithm. A Scatter Search-based approach (SSAR) proposed by Jue et al. [68] is another EA-based FS method. Ant Lion Optimizer (ALO), a recent well-regarded metaheuristic, proposed by S. Mirjalili in [69], was utilized as a searching mechanism in a wrapper FS method in [70,71]. A chaotic ALO approach was proposed for FS in [72]. The GWO, as another recent population-based optimizer [29], has been successfully employed to tackle several applications like the tuning of fuzzy control systems [73]. It has been applied to FS problems [74,75] as well. Recently, a new wrapper-based FS algorithm that uses a hybrid Whale Optimization algorithm (WOA) with SA algorithm as a search method was proposed in [76].

EAs are modeled to mimic the evolution of individuals from their initial states to become better adapted to some objectives imposed upon them. These revolutionary paradigms apply some evolutionary operators (mutation and recombination in GA or pheromone updating rules of ACO) to some selected individuals (based on some selection mechanisms; random, tournament, and roulette wheel selection) in the population to generate an offspring. However, these operators affect and manipulate individuals rather than the whole population. Evolutionary Population Dynamics (EPD) is another evolutionary operator that manipulates the whole population rather than manipulating individuals [77]. Using this operator with EAs will omit the worst individuals from the population rather than improving the best individuals in the population (e.g., recombination in GA). Extremal optimization (EO) [78] is a metaheuristic algorithm that works based on the idea of EPD. The EO algorithm has been used in many research fields with much success [79–81]. The EPD operator is the main feature that enhanced the performance of this algorithm [28].

This paper presents an efficient GOA-based optimizer with EPD and selection operators to improve the efficacy of the basic GOA in dealing with FS tasks. In this work, we have made the following key contributions:

Table 1
List of abbreviations.

Abbreviations	Expansions
Acc	Accuracy
ACO	Ant Colony Optimization
ALO	Ant Lion Optimizer
Atts	Attributes
BGOA	Binary GOA
BGWO	Binary GWO
CFS	Correlation-based Feature Selection
CGFSO	Chaos-based Genetic FS Method
CM	Crossover and Mutation
DFSC	Self-representation Based Dual-graph Regularized Clustering
EA	Evolutionary Algorithms
EO	Extremal Optimization
EPD	Evolutionary Population Dynamics
EPSCO	Evolutionary Programming using Self-Organizing Criticality
FA	Firefly Algorithm
FCBF	Fast Correlation-based Filter
FS	Feature Selection
F-score	Fisher Score
GOA	Grasshopper Optimization Algorithm
GWO	Grey Wolf Optimizer
GA	Genetic Algorithms
IG	Information Gain
k-NN	k-Nearest Neighbor
LO	Local Optima
NSSRD	Non-negative Spectral Learning and Sparse Regression-based Dual-graph Regularized
PSO	Particle Swarm Optimization
RWS	Roulette Wheel Selection
SA	Simulated Annealing
SGFS	Subspace Learning-based Graph Regularized
SOC	Self-Organized Criticality
SSAR	Scatter Search-based Approach
StdDev	Standard Deviation
TS	Tournament Selection
Ts	Tournament Size
WOA	Whale Optimization Algorithm

- The significant merits of the EPD operator motivated our attempts to apply it to the recently proposed Grasshopper Optimization Algorithm (GOA) and investigate its effectiveness on FS problems.
- Four variants of GOA with EPD operator are proposed. In the first two approaches, one of the top three solutions and a randomly generated solution are selected to reposition a solution from the worst half of the population. In the third and fourth approaches, and to give a chance to the low fitness solutions to reformulate the population, two different selection mechanisms (namely Roulette Wheel Selection (RWS) and Tournament Selection (TS) are utilized to select the guiding solution from the first half.
- The proposed approaches have been tested on 22 real benchmarks datasets to show its efficiency for feature selection tasks.
- The hybrid GOA and EPD operator is proposed for the first time to solve the feature selection tasks.
- The proposed GOA based approaches have been tested on real datasets with different settings and characteristics to demonstrate its effectiveness and quality of solutions.

The rest of this paper is organized as follows: [Section 2](#) presents a background about EPD operator. The basics of the GOA algorithm and the hybridization with EPD operator is given in [Section 2](#) as well. [Section 3](#) presents the details of the proposed approaches. In [Section 4](#), the experimental results and analysis are presented. Finally, in [Section 5](#), conclusions and future works are given. [Table 1](#) describes all the abbreviations used in this paper.

2. EPD for the GOA

2.1. Evolutionary Population Dynamics (EPD)

EAs are known as stochastic search methods in which a set of solutions (population) is initialized and then gradually improved to become better adapted to the objectives imposed upon them. Some EAs utilize mutation mechanisms to alter the selected solutions, while others employ the crossover operators. These operators aim to evolve the top selected solutions that are mostly the best solutions. The EPD is the process of eliminating the worst solutions in a population by repositioning them around the best ones. The EPD is basically based on the theory of self-organized criticality (SOC) [82], which indicates that a local change in the population may affect the whole population and provide delicate balances without external organising force [77]. In the GA, the best solutions are combined using the evolutionary operators (crossover and mutation). In contrast, in the EPD, the worst solutions should be omitted from the current population. Evolutionary programming using self-organizing criticality (EPSCO) [83] and Extremal optimization (EO) [78] are two metaheuristics methods that were proposed based on the SCO concept. The EPD is a simple and effective mechanism that can be embedded in different optimizers. It starts by removing the worst solutions from the swarm and then repositioning the removed solutions around the best search agents.

2.2. Grasshopper Optimization Algorithm (GOA)

The GOA is a recent swarm-based nature-inspired algorithm [49] proposed by Saremi et al. It mimics the idealized swarming behavior of grasshopper insects in nature. Similarly other population-based algorithms [84,85], in GOA, a set of candidate

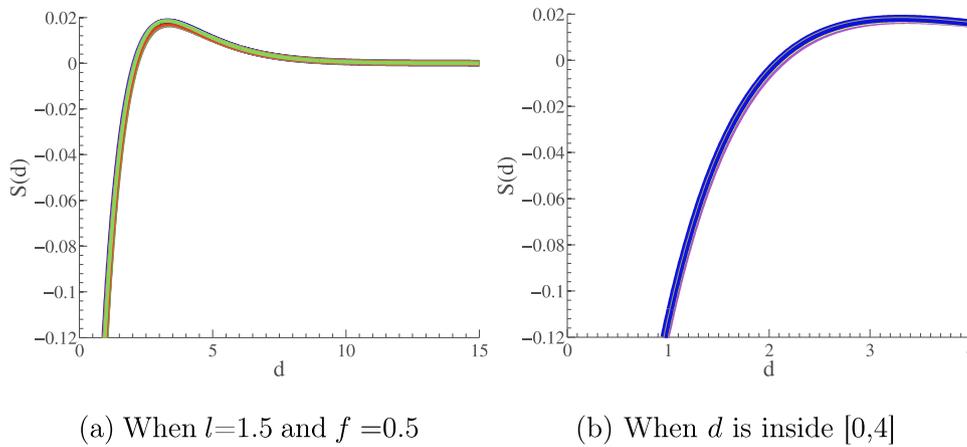


Fig. 1. Function s when $l = 1.5$ and $f = 0.5$ and closer window when d changes in $[0,4]$.

solutions (each individual represents a grasshopper) are randomly generated to construct the initial artificial swarm. Next, all candidate agents are evaluated with regard to the fitness values and the best search agent in the current swarm is considered as the target or leader. The target grasshopper starts attracting the other individuals around its location, and all grasshoppers start moving towards the target grasshopper.

The movement of the i th grasshopper towards the target grasshopper is denoted as X_i and is formulated as in Eq. (1).

$$X_i = S_i + G_i + A_i \tag{1}$$

where S_i is the social interaction, G_i is the gravity force on i th grasshopper, and A_i shows the wind advection. The social interaction S_i acts as the main component during the grasshopper movement process. It can be calculated as Eq. (2):

$$S_i = \sum_{j=1, j \neq i}^N s(d_{ij}) \widehat{d}_{ij} \tag{2}$$

where d_{ij} is the Euclidian distance of the i th with the j th grasshopper, and it is calculated as $d_{ij} = |x_j - x_i|$. While, $\widehat{d}_{ij} = \frac{x_j - x_i}{d_{ij}}$ is a unit vector from the i th grasshopper to the j th grasshopper. The s function is defined as the strength of social forces, which can be calculated as follows:

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \tag{3}$$

where f is the intensity of attraction and l is the attractive length scale.

Fig. 1 illustrates the impact of s -function on the attraction and repulsion (i.e., social interaction) of the grasshoppers. In this figure, the distance d has been considered in the interval of $[0, 15]$. The repulsion force between grasshoppers occurs when the distance between them is between 0 and 2.079 units. In the case that the distance between a grasshopper and other agents is 2.079, it enters to the comfort zone, where neither attraction nor repulsion occurs there, while the attraction starts increasing after 2.079 till 4 and then starts decreasing.

Fig. 1 shows that while the distance between grasshoppers becomes larger, s -function returns values close to 0. Thus, for large distances between grasshoppers, s -function is not capable of applying strong forces to them. To overcome this drawback, the distance between agents are mapped between 1 and 4. The shape of the s -function in the interval $[1, 4]$ is shown in Fig. 1 (right).

Different social behaviors can be obtained for the artificial grasshoppers by changing the parameters l and f of s -function in Eq. (3) as shown in Fig. 2.

The conceptual model of the comfort zone and the attraction and repulsion forces between the grasshoppers is also shown in Fig. 3. The gravity force G_i (second component in Eq. (4)) is calculated as follows:

$$G_i = -g \times \widehat{e}_g \tag{4}$$

where g denotes the gravitational constant and \widehat{e}_g is a unity vector in the vertical direction of the surface.

The wind advection A_i (third component in Eq. (5)) is calculated as follows:

$$A_i = u \times \widehat{e}_w \tag{5}$$

where u represents a constant drift and \widehat{e}_w denotes a unity vector in accordance with the wind.

In stochastic optimisation, a metaheuristic optimizer must make a fine balance between the exploration and exploitation when conducting the search to find a accurate approximation of the global optimum. Therefore, the mathematical formulation of the GOA, which was presented in Eq. (1), should be equipped with special parameters to achieve to this purpose. The mathematical model proposed by Saremi et al. in this regard is as follows:

$$X_i^d = c \left(\sum_{j=1, j \neq i}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) + \widehat{T}_d \tag{6}$$

where ub_d and lb_d are respectively the upper bound and lower bounds in the D th dimension, \widehat{T}_d is the value of the D th dimension of the target grasshopper. Parameter c is a decreasing coefficient to shrink the comfort zone, attraction, and repulsion regions. Note that S is similar to the s function in Eq. (2). In Eq. (6), gravity force has been considered equal to Zero (no G component), and the wind force (A component) is always towards the target grasshopper \widehat{T}_d . The adaptive parameter c is considered as decreasing coefficient, it has been used twice to simulate the deceleration of grasshoppers approaching the source of food and eventually consuming it. The outer c (first c from the left) has been used to reduce the search coverage toward the target grasshopper as the iteration count increases, while the inner c has been used to reduce the effect of the attraction and repulsion forces between grasshoppers with regard to the number of iterations to shrink the comfort, repulsion, and attraction areas.

The parameter c is updated with the following relation, it should be inversely proportional to the number of executed iterations. This mechanism increases the degree of exploitation as the iteration count increases. It also reduces the comfort zone proportional to the number of iterations.

$$c = cMax - l \frac{cMax - cMin}{L} \tag{7}$$

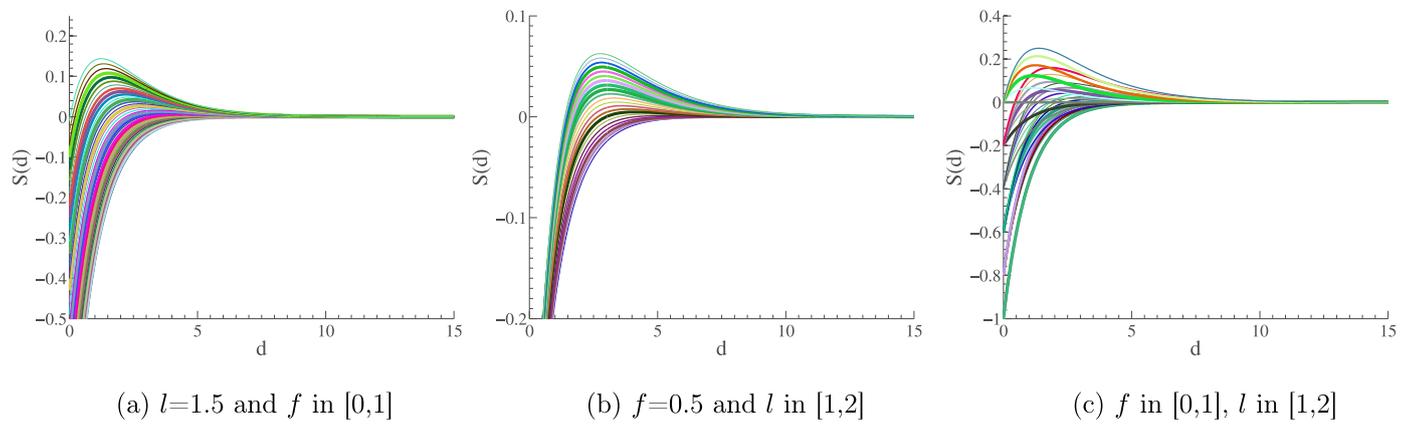


Fig. 2. Behavior of the function s based on l and f .

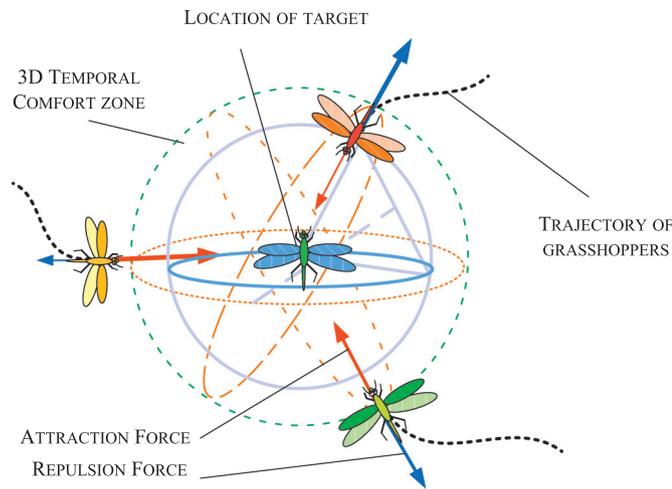


Fig. 3. Primitive corrective patterns between individuals in a swarm of grasshoppers.

where $cMax$ and $cMin$ are respectively the maximum and minimum values of parameter c , l is iteration, and L is the maximum bound of iterations. In [83], the authors used 0.00001 and 1 for $cMin$ and $cMax$, respectively.

3. Binary GOA (BGOA) for feature selection

Finding a minimal feature set has been described as a NP-hard problem [86]. Searching the best combination of features is a challenging problem especially in the wrapper-based methods. Hence, an intelligent optimization method is required to reduce the number of evaluations.

As reported in the literature [53,83], the GOA algorithm can reveal a superior efficacy in tackling various optimization cases. The merits of GOA motivated us to propose a binary version of the GOA optimizer and use it as the core search engine in this paper when solving FS problems [86]. Based on the NP-hard nature of FS problems, where the search space can be represented by binary values, some operators of the GOA algorithm need to be modified. In the continuous GOA, each individual updates its position based on its current position, the position of the best grasshopper found so far (target), and the position of all other grasshoppers as in Eq. (6). This behavior of the GOA is similar to other swarm-based techniques (e.g., PSO). In the GOA, the first term of Eq. (6) is analogous to the velocity vector (step) in the PSO. According to the claims provided by Mirjalili and Lewis [87], one of the easiest ways to

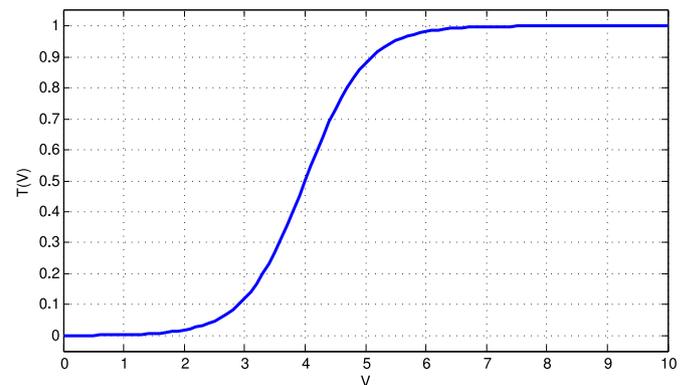


Fig. 4. Sigmoidal Transfer function.

convert an algorithm from continuous to binary version without modifying its structure is to utilize transfer functions. In the proposed approach, the transfer function (see Fig. 4) use the first term in Eq. (6), that is re-defined ΔX in Eq. (8) as the probability for changing of the position elements.

$$\Delta X = c \left(\sum_{j=1, j \neq i}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) \quad (8)$$

Sigmoidal function is a common transfer function proposed by Kennedy and Eberhart [88] as Eq. (9):

$$T(\Delta X_t) = \frac{1}{1 + e^{-\Delta X_t}} \quad (9)$$

where ΔX represents the step vector of a search agent at a specific iteration.

The position of the current grasshopper will be updated as expressed in Eq. (10) based on the probability value $T(\Delta X_t)$ obtained from Eq. (9).

$$X_{t+1}^k(t+1) = \begin{cases} 1 & \text{If } rand < T(\Delta X_{t+1}) \\ 0 & \text{If } rand \geq T(\Delta X_{t+1}) \end{cases} \quad (10)$$

In the wrapper FS, a learning algorithm should be involved in the evaluation of the selected feature subset. In this work, the k -Nearest Neighbor (k -NN) classifier [89] is utilized to attain the classification accuracy of the solution. The higher classification accuracies show that the relevant solution is better. Moreover, since the aim of FS is to eliminate the number of selected features, the smaller the number of features in the solution, the better the solution is. These are two contradictory objectives that should be taken into consideration when designing an objective function for FS al-

gorithms. In this work, the fitness function in Eq. (11) that can balance among the selected features in each agent (minimum) and the accuracy of classification (maximum) is used to evaluate the selected subsets in all approaches.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \quad (11)$$

where $\gamma_R(D)$ is the classification error rate of the known classier, $|R|$ is the number of selected features and $|N|$ is the original number of features, α and β are two parameters to reflect the role of classification rate and length of subset, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ adopted from [70]. The overall pseudocode of the BGOA algorithm is described in Algorithm 1.

Algorithm 1 Pseudocode of the BGOA algorithm.

```

Initialize the GOA parameters cMax, cMin, and maximum iterations L
Initialize a set of random solutions  $X_i (i = 1, 2, \dots, n)$  as initial population
Calculate the fitness of all agents
Remark the best solution as the Target
Set T as the best solution
while  $t < L$  do
  Update c using Eq. (7)
  for each individual in the population do
    Normalize the distances between grasshoppers into [1, 4]
    Update the step vectors ( $\Delta X$ ) using Eq. (8)
    Update position vectors using Eq. (10)
  Update Target if there is a better solution in population
   $t = t + 1$ 
return T

```

3.1. Applying the EPD strategy to BGOA

As discussed earlier, the EPD eliminates the worst solutions from the population and replaces them by generating neighbor solutions around the good ones. The EPD mechanism is a simple but effective operator for population-based techniques [77], therefore, it is applied to the conventional GOA here since it is also a stochastic population-based optimizer. To equip the GOA algorithm with the EPD technique, the swarm of grasshoppers is divided into two parts after sorting it based on the fitness values. The half of the worst grasshoppers is eliminated and reinitialized based on four different strategies depending on the good half of the population.

In this paper, four different strategies are utilized to combine the EPD scheme with the binary BGOA. These versions can be categorized into two main classes based on the implemented selection operators.

3.1.1. BGOA_EP_D with random selection operator

The first model for hybridizing is to use random selection operator. For this purpose, one solution among the best three grasshoppers from the population is selected in addition to a random grasshopper. Then, the leader of 'poor' solution will be selected randomly. To implement this idea, two different approaches are designed that work based on the random selection technique:

1. BGOA_EP_D: it is the simplest hybrid form of the EPD and BGOA algorithms. In this approach, the random selection mechanism is employed to select the solutions. This method also uses a simple mutation operator (see Fig. 5). In this approach, the top three individuals are selected and a fourth solution is generated randomly. Each solution in the worst half is repositioned around any of these four solutions

depending on a random number. The process is straightforward; a random number is generated X_r in each iteration and then one of the following four choices will be applied for repositioning of the poor solution: when $X_r \in [0, 0.25]$, then the best solution is used, when $X_r \in [0.25, 0.5]$, then the second best solution is used, when $X_r \in [0.5, 0.75]$, then the third best solution is employed, and when $X_r \in [0.75, 1]$, a random solution is used.

The selected solution will be used as a starting point to reposition the poor solution. Repositioning the poor solutions around the best solutions aims to heighten the median of the swarm in each step. However, this process may cause a premature convergence of the algorithm. As a remedy, a randomly generated solution is used in the first rule to promote exploration and prevent trapping in local optima.

2. BGOA_EP_D_CM: this version is similar to the BGOA_EP_D and the only difference is that it also uses a crossover and a mutation operator.

In the second approach, a random number is generated and one solution is selected similar to the first strategy, then the selected solution is mutated to improve the exploration tendency of the algorithm. The mutated solution is then crossover with the poor solution.

3.1.2. BGOA_EP_D with special selection operator

According to the findings of Talbi [27], "it does not mean that using better solutions as initial solutions will always lead to better local optima" [27]. The best individuals may bias the searching process and this may cause a premature convergence and a loss of diversity. For this reason, instead of selecting one of the three best solutions like the previous versions, two well-regarded selection mechanisms are applied to select a solution from the first half of the population. For each solution in the second half of the population, select a solution from the first half using the selection mechanism; mutate and then crossover it with the poor solution. In this regard, the alternative method for hybridizing the EPD with the BGOA in the BGOA-EPD algorithm is to also employ a special selection mechanism. There are two well-known selection techniques: Roulette wheel selection (RWS) [90] and the TS [91]. These methods are utilized in this work. Therefore, another two different strategies that can be developed for the BGOA_EP_D are:

1. BGOA_EP_D_Tour: in this version, a solution from the first half of the swarm is selected using the TS operator, then the same crossover and mutation operators utilized in the BGOA_EP_D version are applied on the obtained solution.

The TS is the most popular selection mechanism used with GA due to its efficiency and simple implementation. In TS, a set of n individuals are randomly selected from the whole population, then the best individual among the selected individuals will be selected to reposition the poor solution. The number of selected individuals called tournament size T_s . The advantage of T_s is that it gives a chance to all individuals to guide the poor solutions, which preserve the diversity of the BGOA_EP_D_Tour algorithm. An example of the TS mechanism for the BGOA_EP_D_Tour is illustrated in Fig. 6, where three individuals are selected ($T_s = 3$) and the best solution among them is picked out.

After applying the TS operator, the mutation operator with suitable mutation rate is applied the selected grasshopper hoping to find a better solution in the neighbor of the selected solution and to avoid the BGOA_EP_D_Tour algorithm from the premature convergence. After that, the poor solution is repositioned around the resulting solution by applying a crossover operator. In the BGOA_EP_D_Tour, a solution is selected using the TS to

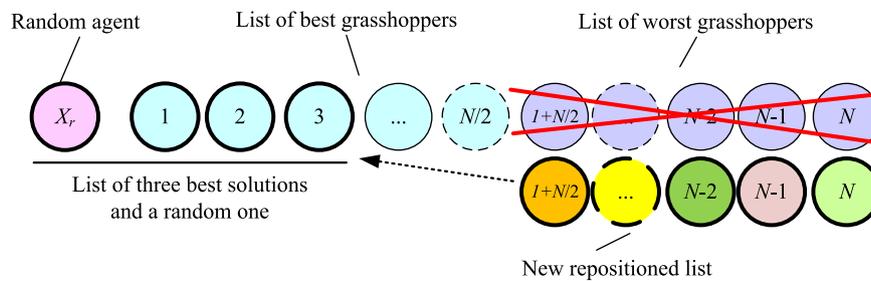


Fig. 5. The sorted population and related agents that should be repositioned around the best solutions and a random one.

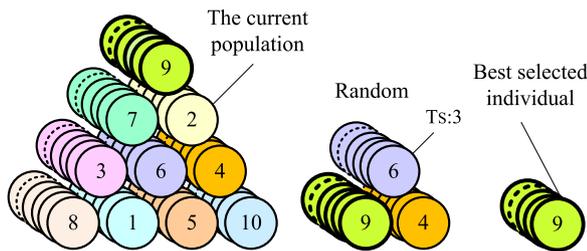


Fig. 6. The mechanism of TS.

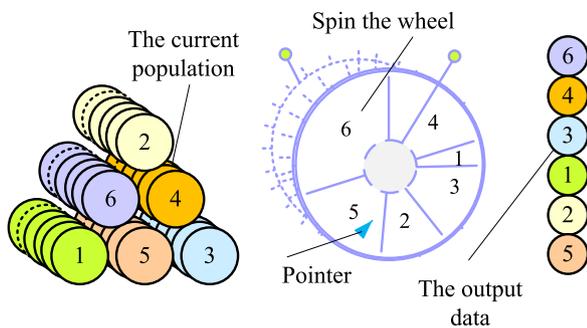


Fig. 7. Selection strategy with RWS.

give a chance to the lower fitness solution in the first half of the population to be selected.

2. BGOA_EPD_RWS: This version is similar to the BGOA_EPD_Tour version and the only difference is that it uses the RWS operator instead of the TS operator.

In the BGOA_EPD_RWS, in each iteration of the BGOA process, for each solution in the worst half of the population, a solution from the first half is selected using the RWS operator. In RWS, individuals are selected with a probability based on their fitness values. In this selection strategy, a roulette wheel is formulated with a circumference equals the sum of all fitness values of the individuals (see Fig 7).

Each individual will have a segment with a size proportional to its fitness. The probability to select an individual can be seen as spinning a roulette wheel, and the segment where the pointer stops is taken and the corresponding individual will be selected. Obviously, the individuals with the largest fitness (i.e. largest segment sizes) have higher probability of being selected than those who have lower probability (i.e. smallest segment sizes). The advantage of RWS that it does not ignore any individual in the population, therefore, it preserves the diversity of the population.

After selecting a solution using the RWS operator, it is mutated to explore more regions of the feature space, then; the resulted solution from the mutation operator is used to reposition a solution from the second half by applying a crossover operator.

The mutation rate r for all related approaches is shown in Eq. (12). The parameter r is decremented from 0.9 to 0, linearly, according to the iteration number i .

$$r = 0.9 \left(1 + \frac{(1-i)}{L-1} \right) \quad (12)$$

where L was the maximum number of iterations.

The main difference between these versions is that they use different selection operators. In addition, the BGOA_EPD_CM uses the best solutions in the population while BGOA_EPD_RWS and BGOA_EPD_Tour variants do not use this policy. They use other solutions from the first half of the population.

The overall pseudo code of the BGOA_EPD algorithm is described in Algorithm 2. Flowchart of the BGOA_EPD is also demon-

Algorithm 2 Pseudo code of the BGOA_EPD approaches.

```

Initialize GOA parameters ( $cMax$ ,  $cMin$ , and  $L$ )
Initialize a set of random solutions  $X_i (i = 1, 2, \dots, n)$  as initial population
Obtain the fitness of all agents
Remark the best solution as Target
while  $t < L$  do
    Update  $c$  using Eq. (7)
    for each individual in the population do
        Normalize the distances between grasshoppers into  $[1, 4]$ 
        Update the step vectors ( $\Delta X$ ) using Eq. (8)
        Update position vectors using Eq. (10)
    Update Target if there is a better solution in the population
    Sort the population based on the fitness
    for  $i = (n/2) + 1$  to  $n$  do
        Update the position of  $i$ -th grasshopper using EPD approach
     $t = t + 1$ 
return Target
    
```

strated in Fig. 8.

Note that the computational complexity of the proposed BGOA_EPD is not significantly different from the GOA. The computational complexity of GOA is of $O(t \times d \times n^2)$ where t indicates the number of iterations, d is the number of variables, and n shows the number of solutions. The proposed binary operators do not change the computational complexity since they have been applied to the position updating mechanism of the original GOA. To re-initialize 50% of solutions, however, the additional complexity of $O(n/2)$ is required, so the overall computational complexity of the proposed BGOA_EPD is $O(t \times d \times n^2 + n/2)$. Note that due to the need to re-evaluate the objective value of half of the solutions, the number of function evaluations in BGOA_EPD is $n/2$ units more than that of GOA.

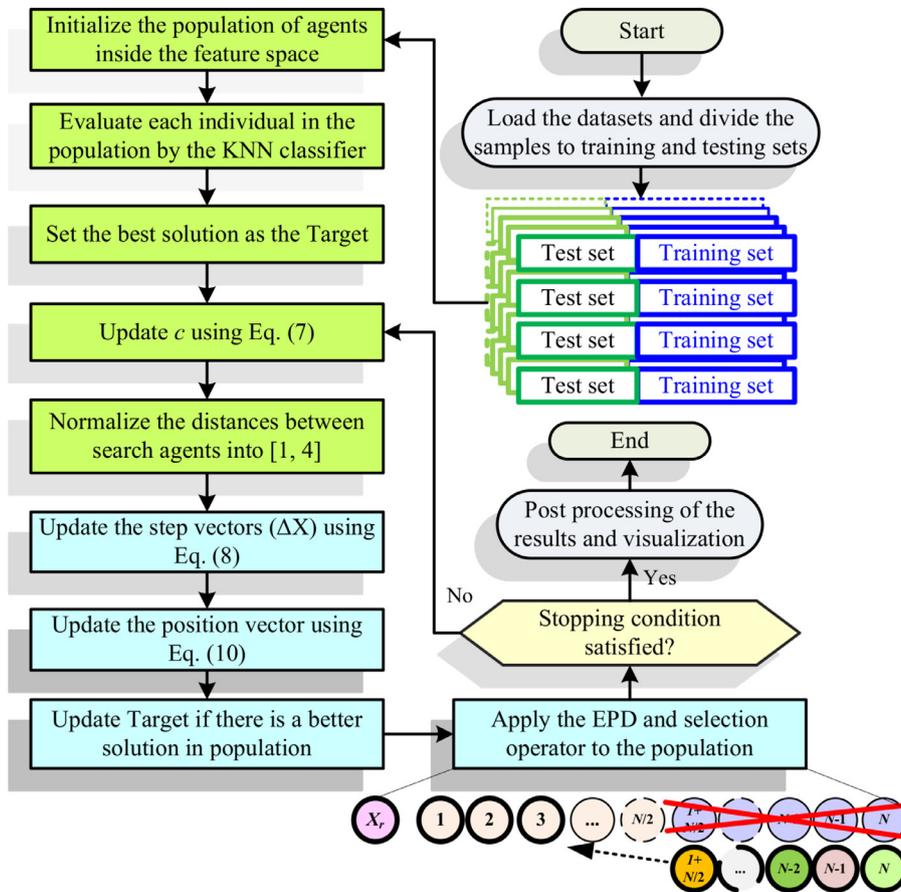


Fig. 8. The overall steps for proposed BGOA_EPDA approach.

4. Experimental results and discussions

In this section, the efficacy of the proposed hybrid EPD_BGOA versions in dealing with 22 well-regarded datasets with different characteristics. Table 2 reports the brief description of the 22 datasets utilized. For more details, interested readers are referred to the UCI [92] repository. These benchmark cases have been stud-

ied in several well-established works. The utilized test set cover different traits and the instances of small to high dimensional datasets and can examine the searching competencies of EA and metaheuristics in tackling the FS problems. Different variants of BGOA algorithm were employed to search for the best reduct with the minimum error rate based on KNN classifier (where $K = 5$ [93]) with the Euclidean distance metric. For evaluation purposes, each datasets is divided into training and testing sets where 80% of the instances in the datasets were used for training purposes and the rest of them is utilized for testing tasks [94].

All the fair tests and the computed results in this research are conducted and prepared on a PC with Intel Core(TM) i5-5200U 2.2GHz CPU and 4.0GB RAM. The maximum iterations (L) is set to 100 and the number of search agents (N) is 10. Additionally, all statistical results are recorded over 30 independent runs. The dimension of cases is equal to the number of features in each experimented dataset. The α and β parameters in the fitness equation are set to 0.99 and 0.01, respectively.

4.1. Evaluation of proposed methods

In this part, the efficiency, convergence and the quality of the results of four developed hybrid approaches are deeply measured and compared to each other's to distinguish the preeminent variant for more advanced investigations. The four techniques utilizing different operators and the random, TS and RWS mechanisms are substantiated and compared to judge and discover the influence of the crossover and mutation strategies and using a specific selection scheme in preference to the random selection policy on either the results or efficacy of the proposed variants. The performance of the proposed optimizers is evaluated and compared in terms of the

Table 2
List of used datasets.

No.	Dataset	No. of Features	No. of instances
1.	Breastcancer	9	699
2.	BreastEW	30	596
3.	Exactly	13	1000
4.	Exactly2	13	1000
5.	HeartEW	13	270
6.	Lymphography	18	148
7.	M-of-n	13	1000
8.	PenglungEW	325	73
9.	SonarEW	60	208
10.	SpectEW	22	267
11.	CongressEW	16	435
12.	IonosphereEW	34	351
13.	KrvskpEW	36	3196
14.	Tic-tac-toe	9	958
15.	Vote	16	300
16.	WaveformEW	40	5000
17.	WineEW	13	178
18.	Zoo	16	101
19.	Clean1	166	476
20.	Semeion	265	1593
21.	Colon	2000	62
22.	Leukemia	7129	72

Table 3
Comparison of the BGOA_S with four hybrid versions using Acc and StdDev metrics .

Dataset	BGOA_S		BGOA_EPD		BGOA_EPD_CM		BGOA_EPD_RWS		BGOA_EPD_Tour	
	Acc	StdDev	Acc	StdDev	Acc	StdDev	Acc	StdDev	Acc	StdDev
Breastcancer	0.969	0.000	0.966	0.000	0.969	0.000	0.977	0.000	0.980	0.001
BreastEW	0.960	0.005	0.962	0.002	0.963	0.003	0.964	0.003	0.947	0.005
Exactly	0.946	0.036	0.997	0.006	0.993	0.009	0.999	0.008	0.999	0.005
Exactly2	0.760	0.000	0.734	0.005	0.744	0.013	0.762	0.000	0.780	0.000
HeartEW	0.826	0.010	0.842	0.006	0.841	0.012	0.815	0.008	0.833	0.004
Lymphography	0.815	0.012	0.895	0.013	0.844	0.012	0.878	0.009	0.868	0.011
M-of-n	0.979	0.030	0.997	0.007	0.999	0.004	0.999	0.004	1.000	0.000
PenglungEW	0.861	0.015	0.868	0.008	0.839	0.021	0.750	0.012	0.927	0.013
SonarEW	0.895	0.011	0.921	0.007	0.883	0.006	0.922	0.008	0.912	0.009
SpectEW	0.851	0.011	0.850	0.007	0.882	0.006	0.852	0.007	0.826	0.010
CongressEW	0.953	0.004	0.983	0.003	0.974	0.004	0.983	0.004	0.964	0.005
IonosphereEW	0.883	0.007	0.907	0.004	0.922	0.007	0.911	0.005	0.899	0.007
KrvskpEW	0.956	0.008	0.960	0.006	0.959	0.006	0.963	0.004	0.968	0.003
Tic-tac-toe	0.803	0.007	0.797	0.001	0.789	0.000	0.785	0.000	0.808	0.000
Vote	0.951	0.004	0.948	0.006	0.953	0.004	0.960	0.003	0.966	0.003
WaveformEW	0.729	0.009	0.739	0.006	0.743	0.004	0.740	0.004	0.737	0.003
WineEW	0.979	0.004	0.994	0.006	0.993	0.005	0.988	0.003	0.989	0.000
Zoo	0.990	0.010	0.976	0.008	0.960	0.004	1.000	0.000	0.993	0.009
Clean1	0.883	0.008	0.885	0.005	0.893	0.005	0.885	0.006	0.863	0.004
Semeion	0.975	0.002	0.979	0.001	0.986	0.001	0.974	0.001	0.976	0.002
Colon	0.745	0.010	0.812	0.012	0.712	0.008	0.810	0.010	0.870	0.006
Leukemia	0.928	0.014	0.889	0.000	0.931	0.014	0.855	0.012	0.931	0.014

Table 4
Average selected attributes using the developed algorithms .

Dataset	BGOA_S		BGOA_EPD		BGOA_EPD_CM		BGOA_EPD_RWS		BGOA_EPD_Tour	
	Atts	StdDev	Atts	StdDev	Atts	StdDev	Atts	StdDev	Atts	StdDev
Breastcancer	4.00	0.000	5.13	0.346	4.17	0.379	4.73	0.980	5.00	0.000
BreastEW	15.37	2.697	20.00	2.729	17.50	1.889	17.20	2.747	17.33	2.440
Exactly	7.63	0.809	6.60	0.498	6.57	0.504	6.43	0.568	6.53	0.571
Exactly2	1.27	0.450	7.97	0.809	5.67	3.565	1.50	0.509	1.53	0.507
HeartEW	6.77	1.524	6.67	0.922	5.77	0.817	6.13	1.548	8.40	1.037
Lymphography	7.47	2.080	10.60	1.003	10.60	1.522	11.60	1.476	10.63	1.217
M-of-n	7.53	0.973	6.53	0.629	6.57	0.504	6.57	0.568	6.47	0.507
PenglungEW	150.13	8.509	166.53	15.937	198.90	8.707	174.77	15.460	178.33	15.486
SonarEW	28.57	3.191	36.37	3.157	36.03	3.489	35.73	2.778	36.77	4.240
SpectEW	9.93	1.856	14.13	1.995	12.83	2.437	11.93	2.463	11.10	3.044
CongressEW	4.33	1.322	7.67	1.729	6.53	2.145	5.50	1.592	5.77	2.012
IonosphereEW	13.43	3.115	18.93	3.269	17.93	2.420	17.77	3.785	16.40	3.701
KrvskpEW	19.90	3.010	22.33	3.010	22.10	2.551	22.43	2.417	21.67	2.496
Tic-tac-toe	6.83	0.379	5.03	0.183	6.00	0.000	6.10	0.305	5.00	0.000
Vote	5.27	2.083	6.73	1.015	6.57	1.995	6.07	1.617	5.43	1.223
WaveformEW	21.20	2.952	25.53	3.082	25.80	3.295	24.93	2.876	26.23	3.451
WineEW	6.33	1.348	7.27	0.944	7.23	0.817	6.83	1.147	8.80	1.472
Zoo	8.13	1.167	7.97	1.426	7.93	1.230	7.77	0.774	9.17	1.967
clean1	82.93	5.948	103.80	6.880	105.20	6.206	96.00	9.127	92.60	7.802
Semeion	134.90	7.662	169.63	11.137	171.50	7.361	159.87	9.885	157.03	11.485
Colon	967.87	23.882	1194.33	89.430	1198.33	97.070	1050.83	70.247	1063.67	64.618
Leukemia	3495.23	48.699	4138.27	373.767	4366.87	288.129	3896.63	292.866	3768.80	224.842

average classification accuracy (Acc), selection size, and fitness values, computational times, P-values of the Wilcoxon ranksum test and convergence rates over all runs of each technique. The Acc is measured via the nominated features on the used dataset. The standard deviation (StdDev) of all proposed versions is also provided for all metrics, datasets and algorithms.

Table 3 exposes the attained Acc and related StdDev results for the BGOA_S algorithm versus other designed versions. Tables 4–6 also reflect the average selected attributes, fitness, and CPU time values along with the related StdDev for the proposed techniques.

From Table 3, it can be detected that the hybrid BGOA_EPD_Tour can relatively outperform other competitors in terms of Acc and StdDev metrics in dealing with 10 and 11 datasets, respectively. The simple binary BGOA_S cannot reveal higher accuracies than any hybrid variant over all 22 datasets. For the M-of-n dataset, the BGOA_EPD_Tour has classified with

100% Acc and 0 StdDev using only 6.47 attributes. From Table 3, it is observed that the BGOA_EPD_RWS can provide superior Acc rates compared to other varieties in tackling the BreastEW, CongressEW, SonarEW, and Zoo datasets. The BGOA_EPD_RWS has attained the Acc of 100% in solving the Zoo test case. The BGOA_EPD_CM outperform others in terms of Acc in dealing with the 6 datasets: IonosphereEW, SpectEW, WaveformEW, Clean1, Semeion, and Leukemia problems. The BGOA_EPD has outperformed competitors in realizing the HeartEW, Lymphography, and WineEW. In tackling the CongressEW, both BGOA_EPD and BGOA_EPD_RWS have reached to a same Acc rate 98.3%, while based on selected attributes in Table 4, the BGOA_EPD_RWS with 5.5 selected attributes has outperformed the BGOA_EPD.

Regarding the Acc rates, the BGOA_EPD_Tour can outperform the BGOA_S over 19 problems and there is a marked shift in the rates for the BGOA_EPD_Tour and improvement varies from 1%

Table 5
Average fitness values for proposed versions.

Dataset	BGOA_S		BGOA_EPD		BGOA_EPD_CM		BGOA_EPD_RWS		BGOA_EPD_Tour	
	Fitness	StdDev	Fitness	StdDev	Fitness	StdDev	Fitness	StdDev	Fitness	StdDev
Breastcancer	0.036	0.000	0.040	0.000	0.036	0.000	0.028	0.001	0.026	0.001
BreastEW	0.045	0.005	0.044	0.002	0.043	0.003	0.041	0.003	0.058	0.004
Exactly	0.059	0.036	0.008	0.006	0.012	0.010	0.006	0.008	0.006	0.006
Exactly2	0.239	0.000	0.269	0.005	0.257	0.015	0.237	0.000	0.219	0.000
HeartEW	0.178	0.009	0.161	0.006	0.162	0.012	0.188	0.008	0.171	0.004
Lymphography	0.187	0.012	0.110	0.013	0.161	0.012	0.127	0.009	0.137	0.011
M-of-n	0.027	0.030	0.008	0.007	0.006	0.004	0.006	0.004	0.005	0.000
PenglungEW	0.142	0.015	0.136	0.008	0.166	0.020	0.253	0.012	0.078	0.012
SonarEW	0.109	0.010	0.084	0.007	0.122	0.006	0.083	0.008	0.094	0.008
SpectEW	0.152	0.011	0.155	0.007	0.123	0.006	0.152	0.006	0.177	0.010
Tic-tac-toe	0.203	0.007	0.206	0.001	0.215	0.000	0.220	0.000	0.196	0.000
CongressEW	0.049	0.004	0.022	0.003	0.030	0.004	0.020	0.005	0.039	0.005
IonosphereEW	0.120	0.008	0.098	0.004	0.082	0.007	0.094	0.005	0.105	0.007
KrvskpEW	0.049	0.008	0.046	0.006	0.047	0.006	0.043	0.004	0.038	0.003
Vote	0.052	0.004	0.055	0.006	0.050	0.005	0.044	0.003	0.037	0.003
WaveformEW	0.274	0.009	0.265	0.006	0.261	0.004	0.263	0.005	0.267	0.003
WineEW	0.025	0.004	0.012	0.005	0.013	0.005	0.017	0.003	0.018	0.001
Zoo	0.015	0.010	0.028	0.007	0.044	0.004	0.005	0.000	0.012	0.008
Clean1	0.121	0.008	0.120	0.004	0.112	0.005	0.120	0.006	0.141	0.004
Semeion	0.030	0.002	0.027	0.001	0.020	0.001	0.031	0.001	0.030	0.001
Colon	0.257	0.010	0.192	0.012	0.291	0.008	0.194	0.010	0.134	0.006
Leukemia	0.076	0.014	0.116	0.001	0.074	0.014	0.149	0.012	0.073	0.014

Table 6
Average CPU time (seconds) of proposed techniques.

Dataset	BGOA_S		BGOA_EPD		BGOA_EPD_CM		BGOA_EPD_RWS		BGOA_EPD_Tour	
	Time	StdDev	Time	StdDev	Time	StdDev	Time	StdDev	Time	StdDev
Breastcancer	3.537	0.248	4.053	0.232	4.370	0.222	4.131	0.214	4.146	0.191
BreastEW	3.780	0.199	5.256	0.221	5.761	0.237	5.427	0.249	5.404	0.241
Exactly	4.874	0.238	6.166	0.301	6.881	0.313	6.155	0.262	6.014	0.277
Exactly2	5.223	0.344	6.204	0.285	6.882	0.351	5.934	0.265	5.942	0.257
HeartEW	2.803	0.164	3.276	0.183	3.667	0.165	3.484	0.190	3.505	0.178
Lymphography	2.540	0.148	3.145	0.144	3.547	0.169	3.413	0.161	3.427	0.162
M-of-n	4.792	0.234	5.641	0.220	6.406	0.288	5.812	0.279	6.173	0.271
penglungEW	2.836	0.164	18.597	0.784	21.038	0.920	19.494	0.756	19.560	0.822
SonarEW	2.713	0.160	5.469	0.263	6.113	0.262	5.774	0.240	5.801	0.245
SpectEW	2.705	0.154	3.465	0.147	3.916	0.189	3.746	0.183	3.775	0.164
Tic-tac-toe	4.226	0.210	4.961	0.210	5.227	0.255	5.214	0.250	5.097	0.222
CongressEW	3.248	0.160	3.835	0.186	4.273	0.207	4.093	0.211	4.109	0.205
IonosphereEW	2.899	0.152	4.441	0.205	4.988	0.207	4.717	0.201	4.735	0.199
KrvskpEW	49.520	1.668	58.357	1.773	62.814	3.055	56.831	1.871	57.043	1.743
Vote	2.799	0.158	3.364	0.164	3.491	0.154	3.618	0.187	3.640	0.179
WaveformEW	123.546	4.381	152.292	4.186	157.271	5.641	145.890	4.420	152.167	4.812
WineEW	2.492	0.151	3.054	0.152	3.176	0.152	3.281	0.176	3.766	0.301
Zoo	2.485	0.145	3.076	0.137	3.182	0.156	3.371	0.186	3.614	0.240
clean1	7.204	0.225	16.423	0.542	16.988	0.566	16.904	0.600	17.220	0.608
Semeion	88.934	1.514	129.229	2.182	130.251	2.154	120.308	2.480	123.254	2.147
Colon	4.964	0.338	100.832	3.628	106.310	4.461	107.901	4.333	112.236	4.364
Leukemia	15.322	0.884	383.067	19.075	379.074	15.501	381.100	15.244	394.688	15.418

to 12.5%. A comparable pattern can be detected from the results of BGOA_EPD_RWS, it outperforms the BGOA_S technique on 17 datasets, whereas the BGOA_EPD_CM and BGOA_EPD outperform the basic optimizer on 15 problems.

According to the selected attributes (Atts) in Table 4, it is seen that the simple BGOA_S is better than the BGOA_EPD_Tour on 19 datasets. It is also superior to the BGOA_EPD_RWS and BGOA_EPD_CM in dealing with 17 problems.

Inspecting the fitness measures (Fitness) in Table 5, the best optimizer is the BGOA_EPD_Tour. It shows the lowest values for the objective function in tackling the 10 datasets: Breastcancer, Exactly, Exactly2, KrvskpEW, M-of-n, PenglungEW, Tic-tac-toe, Vote, Colon, and Leukemia. The BGOA_EPD_RWS has shown a relatively good performance in dealing with 4 test cases: BreastEW, CongressEW, SonarEW, and Zoo. The BGOA_EPD_CM has provided a lower fit-

ness for IonosphereEW, SpectEW, WaveformEW, Clean1, and Semeion datasets.

From Table 6, it can be observed that the BGOA_S is the fastest approach in a same computational environment with other optimizers. When comparing the algorithms with the TS-based and RWS-based selection operators, it is seen that for 19 datasets, BGOA_EPD_RWS outperforms the BGOA_EPD_Tour. For only BreastEW, Exactly, and Tic-tac-toe, the BGOA_EPD_Tour has a slightly better run speed. In addition, from Table 7, BGOA_EPD_RWS and BGOA_EPD_Tour are confirmed that the observed differences are statistically significant for majority of datasets according to p-values of Wilcoxon ranksum test with 5% significance.

The average ranking of the proposed binary and hybrid versions in terms of Acc, Att, and fitness metrics is presented in Table 8. In Table 8, the rank of each method on a dataset is calculated, then,

Table 7

P-values of the Wilcoxon test the classification accuracy results of the proposed approaches ($p \geq 0.05$ are underlined, and N/A means not applicable).

	BGOA_S	BGOA_EPD	BGOA_EPD-CM	BGOA_EPD_RWS	BGOA_EPD_Tour
Breast Cancer	2.71E-14	2.71E-14	2.71E-14	1.22E-12	N/A
BreastEW	1.42E-04	1.32E-02	<u>2.25E-01</u>	N/A	1.88E-11
Exactly	2.04E-10	9.53E-06	1.21E-04	1.00E+00	N/A
Exactly2	1.69E-14	1.01E-12	5.37E-13	1.69E-14	N/A
HeartEW	7.62E-09	N/A	<u>2.15E-01</u>	1.29E-11	7.19E-08
Lymphography	1.98E-11	N/A	1.79E-11	6.41E-06	1.89E-11
M-of-n	8.60E-07	4.19E-02	<u>8.15E-02</u>	<u>3.34E-01</u>	N/A
penglungEW	3.75E-12	8.38E-13	5.90E-12	2.63E-12	N/A
SonarEW	4.77E-11	<u>7.19E-01</u>	7.57E-12	N/A	2.11E-05
SpectEW	2.46E-11	1.56E-11	N/A	1.29E-11	1.79E-11
CongressEW	1.35E-11	<u>6.58E-01</u>	3.87E-09	N/A	1.54E-11
IonosphereEW	1.82E-11	2.68E-10	N/A	2.95E-08	5.21E-11
KrvskpEW	3.14E-09	4.23E-06	1.48E-08	3.29E-06	N/A
Tic-tac-toe	1.19E-13	2.71E-14	2.71E-14	1.69E-14	N/A
Vote	9.06E-12	7.00E-12	1.64E-11	2.11E-08	N/A
WaveformEW	3.04E-08	3.16E-03	N/A	3.13E-02	3.73E-07
WineEW	1.07E-10	N/A	<u>4.35E-01</u>	2.50E-05	5.59E-05
Zoo	3.80E-06	1.55E-13	2.71E-14	N/A	6.18E-04
Clean1	1.09E-06	4.81E-07	N/A	2.15E-06	1.94E-11
semeion	1.77E-11	1.80E-11	N/A	1.83E-11	1.96E-11
Colon	9.45E-14	2.39E-13	6.50E-14	1.13E-13	N/A
Leukemia	<u>3.09E-01</u>	4.63E-13	N/A	3.31E-12	N/A

Table 8

Overall ranking results.

Algorithm	BGOA_S			BGOA_EPD			BGOA_EPD_CM			BGOA_EPD_RWS			BGOA_EPD_Tour		
	Acc	Att	Fit	Acc	Att	Fit	Acc	Att	Fit	Acc	Att	Fit	Acc	Att	Fit
Breastcancer	3	1	3	5	5	5	3	2	3	2	3	2	1	4	1
BreastEW	4	1	4	3	5	3	2	4	2	1	2	1	5	3	5
Exactly	5	5	5	3	4	3	4	3	4	1	1	1	1	2	1
Exactly2	3	1	3	5	5	5	4	4	4	2	2	2	1	3	1
HeartEW	4	4	4	1	3	1	2	1	2	5	2	5	3	5	3
Lymphography	5	1	5	1	2	1	4	2	4	2	5	2	3	4	3
M-of-n	5	5	5	4	2	4	2	3	2	2	3	2	1	1	1
PenglungEW	3	1	3	2	2	2	4	5	4	5	3	5	1	4	1
SonarEW	4	1	4	2	4	2	5	3	5	1	2	1	3	5	3
SpectEW	3	1	2	4	5	4	1	4	1	2	3	2	5	2	5
CongressEW	5	1	5	1	5	2	3	4	3	1	2	1	4	3	4
IonosphereEW	5	1	5	3	5	3	1	4	1	2	3	2	4	2	4
KrvskpEW	5	1	5	3	4	3	4	3	4	2	5	2	1	2	1
Tic-tac-toe	2	5	2	3	2	3	4	3	4	5	4	5	1	1	1
Vote	4	1	4	5	5	5	3	4	3	2	3	2	1	2	1
WaveformEW	5	1	5	3	3	3	1	4	1	2	2	2	4	5	4
WineEW	5	1	5	1	4	1	2	3	2	4	2	3	3	5	4
Zoo	3	4	3	4	3	4	5	2	5	1	1	1	2	5	2
Clean1	4	1	4	2	4	2	1	5	1	2	3	2	5	2	5
Semeion	4	1	3	2	4	2	1	5	1	5	3	5	3	2	3
Colon	4	1	4	2	4	2	5	5	5	3	2	3	1	3	1
Leukemia	3	1	3	4	4	4	1	5	2	5	3	5	1	2	1
Sum of the ranks	88	40	86	63	84	64	62	78	63	57	59	56	54	67	55
Overall rank	5	1	5	3	5	4	3	4	3	2	2	2	1	3	1
Total sum	214			211			203			172			176		
Final ranks	5			4			3			1			2		

the sum of the ranks based on each metric is obtained. The total sum shows the sum of the ranks of each optimizer based on all metrics. The final rank shows the final average place of each algorithm in handling all 22 datasets. The utilized ranking system gives a lower place to those items that have better value according to a specific metric. The overall and final ranks are inside [1, 5] interval.

Table 8 divulges that the BGOA_EPD_RWS and BGOA_EPD_Tour has achieved to the best places. Regarding the Acc metric, the BGOA_EPD_Tour won the competition and BGOA_EPD_RWS was the second winner, while the BGOA_EPD_CM and BGOA_EPD were both the third front-runners and BGOA_S gained the last stage. Based on the orders for the Fit measure, the best solvers can be spotted as the BGOA_EPD_Tour, BGOA_EPD_RWS, BGOA_EPD_CM, BGOA_EPD, and BGOA_S, respectively. Regarding the Att mea-

sure, the best is the BGOA_S, while the BGOA_EPD_Tour and BGOA_EPD_RWS have acquired the second and third places.

Based on the final ranks, the best two algorithms (BGOA_EPD_RWS and BGOA_EPD_Tour) have demonstrated very competitive performances and the ranks are 172 and 176, respectively. There is a notable gap between the ranks of top winners having the RWS and TS selection schemes and the third front-runner, the BGOA_EPD_CM that runs the mutation and crossover to relocate the feeble grasshoppers based on top front-runners of the population.

The reason might be that the BGOA_EPD_CM and BGOA_EPD have utilized the best grasshoppers of the population, and this can bias the exploration phase, which has instigated premature convergence and a loss of diversity in the population. This effect of best

solutions has decreased the quality of the outcomes, and consequently, the grades of the BGOA_EPD_CM and BGOA_EPD in the ranking system have dropped compared to the BGOA_EPD_Tour and BGOA_EPD_RWS. From the other side, BGOA_EPD_Tour inherits the advantage of TS, which can preserve the diversity of grasshoppers and consequently, the BGOA_EPD_Tour has an advanced potential to retain and recover a stable balance between the exploration and exploitation inclinations. In addition, BGOA_EPD_Tour applies the mutation scheme to the selected grasshopper hoping to find a better solution in the neighbor of the selected solution and to avoid the LO. For these reasons, the BGOA_EPD_Tour can outperform all based on Acc and fitness measures.

Based on final ranks, the best version is the BGOA_EPD_RWS. The reason is that this version does not ignore any grasshopper in the population, so it is capable of preserving the diversity of the agents, which can help the BGOA_EPD_RWS to perform deeper exploration levels. This fact has enabled the BGOA_EPD_RWS to avoid LO and discover better results. In addition, after selecting an agent with the RWS operator, it is mutated to explore more areas of the feature space, then; the resulted solution is used again to reposition a solution from the second half by applying the crossover operator. these operators also improve the efficacy of the BGOA_EPD_RWS in balancing the exploration and exploitation as compared to other developed versions.

The convergence trends for the all proposed variants according to the fitness measure on all 22 datasets are also compared and demonstrated in Figs. 9 and 10.

From Fig. 9, it can be observed that the BGOA_EPD_RWS has exposed the best curves compared to other versions in tackling the BreastEW, CongressEW, Lymphography, and SonarEW problems. The BGOA_EPD_Tour can reveal a quicker tendency than others in treating the Breastcancer, Exactly2, HeartEW, KrvskepEW, M-of-n, and penglungEW datasets, while the BGOA_EPD has stagnated to LO in early steps of the exploration phase, for example when solving the Breastcancer, CongressEW, Exactly, HeartEW, lonosphereEW, KrvskepEW, Lymphography, and penglungEW tasks. From the curves on Fig. 9, it can be inferred that the approaches utilizing the TS and RWS schemes have a head-to-head convergence proclivity.

It seems that the TS and RWS-embedded variants perform better than versions employing random selection and top solutions for rearranging of the worst grasshoppers. This reveals that enhancing the median of all population using only top solutions and a random grasshopper may increase the chance of BGOA_EPD to be easily captivated in LO when penetrating the fruitless regions of the feature space. The trends of the BGOA_EPD_CM also support that the crossover and mutation schemes has heightened the inclusive leaning of the BGOA_EPD in balancing the exploration and exploitation traits. Therefore, it is seen that the BGOA_EPD_CM can converge faster to better results than the BGOA_EPD in basically all cases. And yet, the BGOA_EPD_CM cannot surpass both BGOA_EPD_Tour and BGOA_EPD_RWS in terms of convergence results, except in solving the lonosphereEW. Therefore, it is seen that using selection mechanisms has alleviated the unripe convergence shortcoming of the BGOA_EPD. In RWS-based version, the door is open for the weak solution in the first half of the population to be selected but the better solutions have more chance. In this regard, the BGOA_EPD_Tour attains better results than BGOA_EPD_RWS.

According to Fig. 10, the convergence shortcomings of the BGOA_EPD can still be detected. However, it converged to improved results for the Clean1. The BGOA_S can be better than the BGOA_EPD regarding the convergence and it shows competitive trends for all cases.

The BGOA_EPD_CM has revealed enriched tendencies compared to the BGOA_EPD and BGOA_S, mainly on SpectEW, WaveformEW, WineEW, and Semeion problems, which has slightly out-

Table 9

The parameter settings.

Algorithm	Parameter	Value
GSA	G_0	100
	α	20
BA	Q_{\min} Frequency minimum	0
	Q_{\max} Frequency maximum	2
	A Loudness	0.5
	r Pulse rate	0.5
GWO	a	[2 0]

performed other methods. It shows that the extra operators assist the BGOA_EPD in fleeing from the LO. When the TS-based selection theme has met the BGOA_EPD, it has shown best curves on the Tic-tac-toe, Vote, Colon, and Leukemia, whereas the BGOA_EPD_RWS has not exposed the excellent curves, except on the Zoo. The reason is that the TS can enhance and preserve the diversity of the solutions, which can encourage more stable balance between the local and global search trends. Then, it helps the BGOA_EPD_Tour to uncover superior trends on harder cases like the Leukemia with 7129 and Colon with 2000 features. It is seen that the tendencies of BGOA_EPD_Tour and BGOA_EPD_RWS are very competitive on the WineEW and WaveformEW.

Considering all results, convergence curves and final rankings of algorithms, it can be recognized that two best versions are the BGOA_EPD_RWS and BGOA_EPD_Tour algorithms. In the next section, the BGOA_EPD_Tour, which is the best hybrid variant of BGOA_EPD, is considered to be further compared to other well-established optimizers with regard to the efficacy, performance on different metrics and convergence behaviors.

4.2. Comparison with other metaheuristics

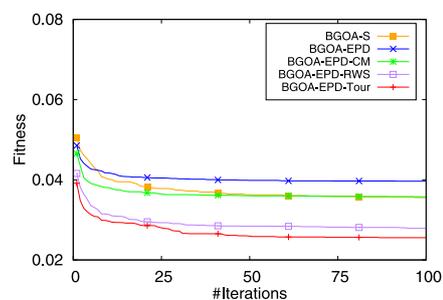
In this section, the efficacy and qualitative results of the BGOA_EPD_Tour techniques is compared to the several well-regarded and related optimizers in the FS field from different aspects. The binary versions of the GWO (bGWO) [74], GSA (BGSA) [95], and BA (BBA) [96] are utilized here to deeply investigate the comparative efficiency of these well-established methods against the developed BGOA-based version.

All trials have completed during a same condition and all conditions were similar to the described information in preceding section. The parameters of optimizers are sensibly selected using many trial and error processes to comprehend the finest feasible settings. Table 9 shows the used parameters.

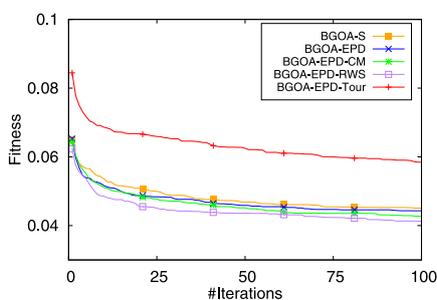
Table 10 exposes the attained Acc and related StdDev results for the proposed algorithms versus other metaheuristics. Tables 11–13 also reflect the average selected attributes, fitness, and CPU time values along with the related StdDev for the compared techniques. Table 14 shows the results of Wilcoxon ranksum statistical test for the accuracy results in Table 10. For the used test, the best technique for each dataset is considered as the base method to be compared with other peers, independently.

From Table 10, it is seen that the hybrid BGOA_EPD_Tour can evidently outperform all contestants on 20 datasets. The bGWO also outperform others on 2 problems: BreastEW and Clean1. In comparison with the BGSA, the BGOA_EPD_Tour can provide better rates on all problems. In dealing with 21 datasets, the classification accuracies of the BGOA_EPD_Tour have improved in the interval of 0.5% (Semeion) to 30% (Exactly) in comparison with the BGSA. It can also outperform the BBA on all 22 datasets.

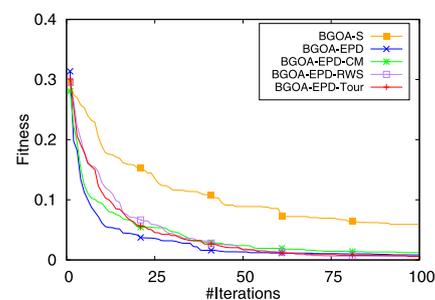
The best and worst average Acc that the bGWO have reached is 97.45% and 66.13% on Zoo and Colon datasets, respectively. For M-of-n case, the BGOA_EPD_Tour have reached to 100% accuracy, while bGWO has the accuracy of 89.41%, which shows the su-



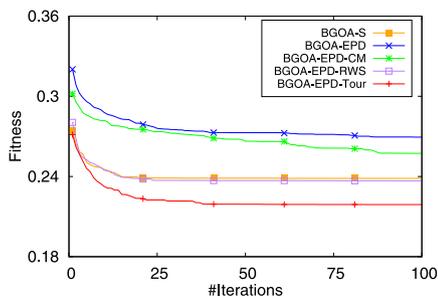
(a) Breastcancer



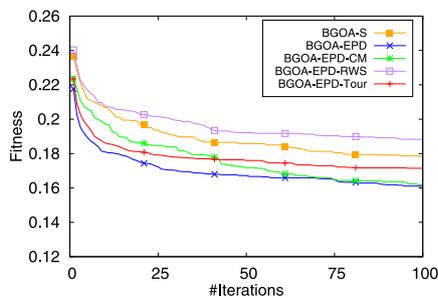
(b) BreastEW



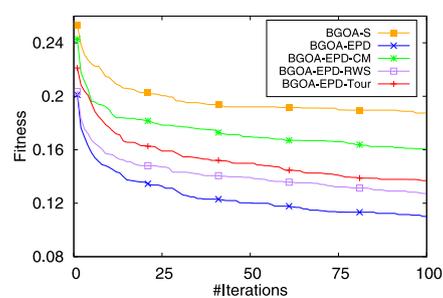
(c) Exactly



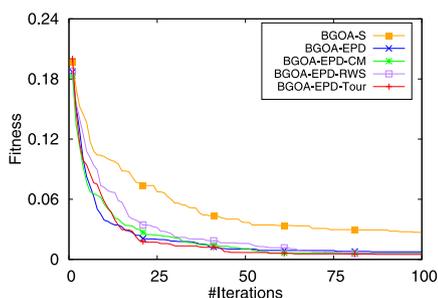
(d) Exactly2



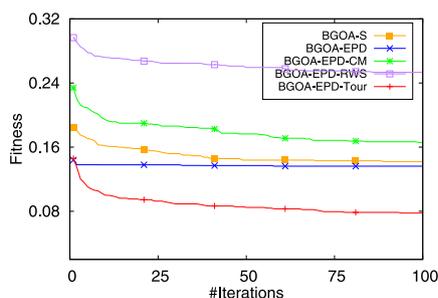
(e) HeartEW



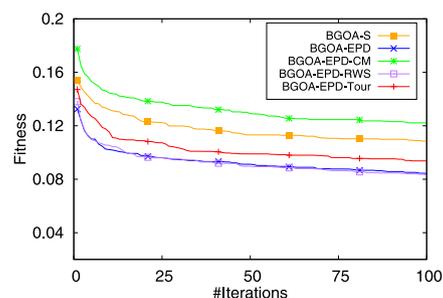
(f) Lymphography



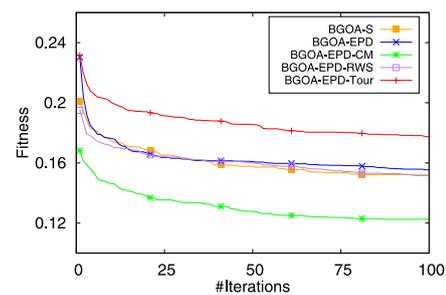
(g) M-of-n



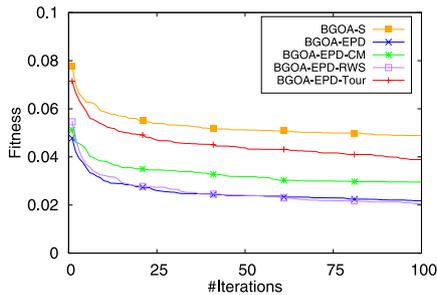
(h) penglungEW



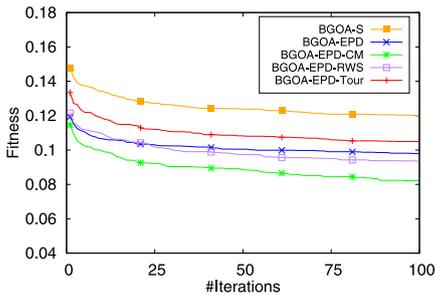
(i) SonarEW



(j) SpectEW



(k) CongressEW



(l) IonosphereEW

Fig. 9. Convergence curves of the proposed approaches for Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, and SonarEW, SpectEW, CongressEW, and IonosphereEW datasets.

perior efficacy of the proposed EPD-based optimizer. Based on the overall ranks, the BGOA_EPDTour has achieved to the first place and bGWO, BGSa, and BBA are the next choices, respectively. Several substantial improvements in the StdDev index can also be detected. The main reason for improved efficacy of the BGOA_EPDTour is that the poor solutions are repositioned around

the better ones in the BGOA_EPDTour using the EPD, and during this process, the selection mechanisms has assisted the proposed approach to maintain the diversity of swarm, and then, recover a fine balance between the exploration and exploitation. Therefore, in the case of stagnation to LO, they can escape from them using the random nature behind the utilized operators.

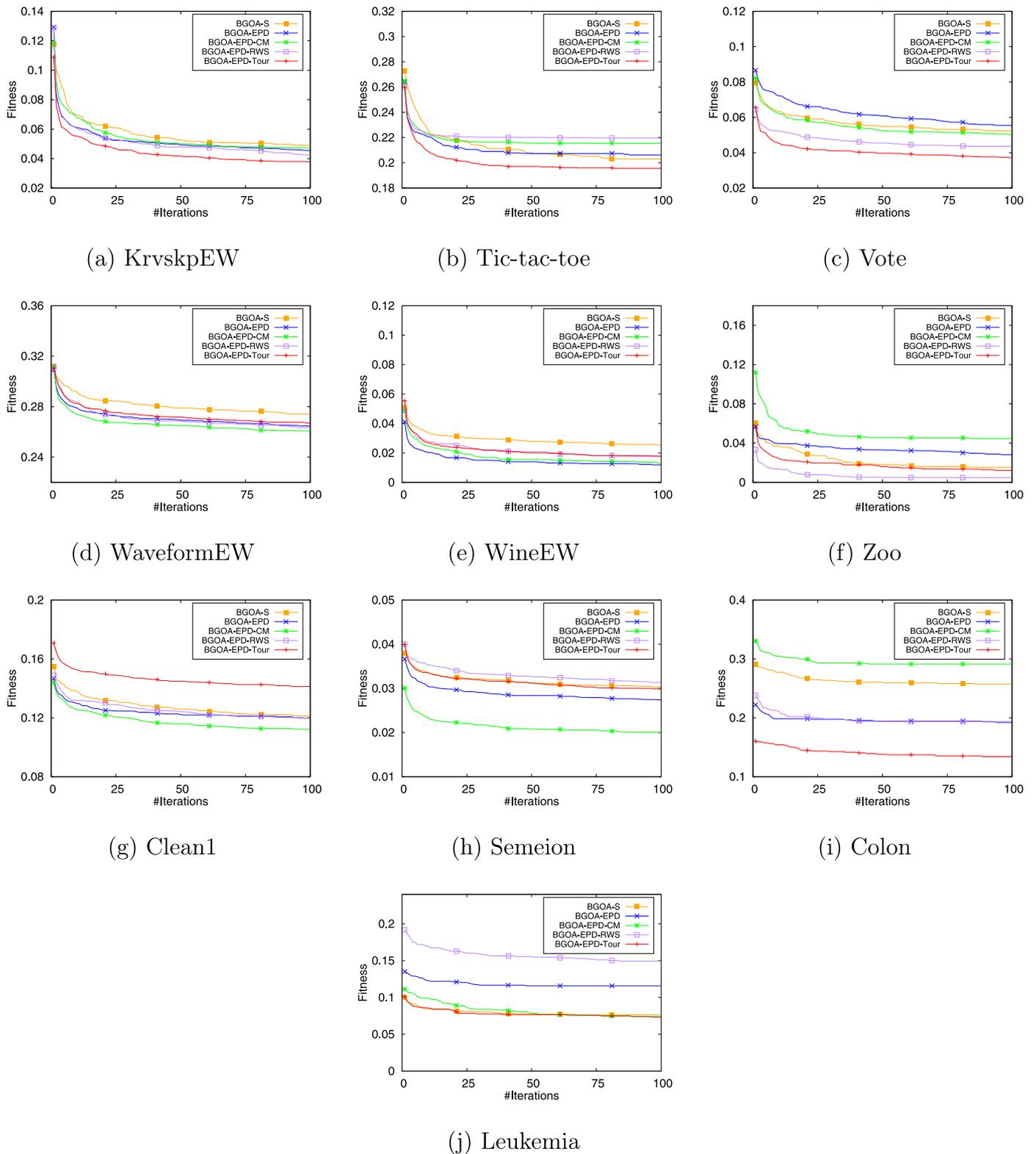


Fig. 10. Convergence curves of the proposed approaches for KrvskepEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Zoo, Clean1, Semeion, Colon, and Leukemia datasets.

From the results reflected in Table 11, it is evident that the BBA algorithm is better than other algorithms on 19 datasets. The BGOA-EPD-Tour and BGSA algorithms have attained the next ranks. For CongressEW and Vote, it is observed that the BGOA-EPD-Tour can show the best results.

Inspecting the results in Table 12, it is observed that the proposed BGOA-EPD-Tour is capable of outperforming all algorithms and revealing the best costs in realizing 20 datasets.

The BGOA-EPD-Tour shows superior costs compared to the BBA, BGSA, and bGWO algorithms on 90.9%, 95.45%, and 90.9% of the datasets, respectively. The reason is that the TS-based selection op-

Table 10
Classification accuracy results of the BGOA_EPD_Tour compared to other metaheuristics.

Dataset	BGOA_EPD_Tour		bGWO		BGSa		BBA	
	Acc	StdDev	Acc	StdDev	Acc	StdDev	Acc	StdDev
Breastcancer	0.980	0.001	0.968	0.002	0.957	0.004	0.937	0.031
BreastEW	0.947	0.005	0.954	0.007	0.942	0.006	0.931	0.014
Exactly	0.999	0.005	0.809	0.076	0.697	0.060	0.610	0.065
Exactly2	0.780	0.000	0.743	0.017	0.706	0.023	0.628	0.057
HeartEW	0.833	0.004	0.792	0.017	0.777	0.022	0.754	0.033
Lymphography	0.868	0.011	0.813	0.028	0.781	0.022	0.701	0.069
M-of-n	1.000	0.000	0.894	0.041	0.835	0.063	0.722	0.080
penglungEW	0.927	0.013	0.850	0.014	0.919	0.000	0.795	0.029
SonarEW	0.912	0.009	0.836	0.016	0.888	0.015	0.844	0.036
SpectEW	0.826	0.010	0.810	0.014	0.783	0.024	0.800	0.027
CongressEW	0.964	0.005	0.948	0.011	0.951	0.008	0.872	0.075
IonosphereEW	0.899	0.007	0.885	0.009	0.881	0.010	0.877	0.019
KrvskpEW	0.968	0.003	0.934	0.015	0.908	0.048	0.816	0.081
Tic-tac-toe	0.808	0.000	0.754	0.032	0.753	0.024	0.665	0.063
Vote	0.966	0.003	0.944	0.010	0.931	0.011	0.851	0.096
WaveformEW	0.737	0.003	0.723	0.007	0.695	0.014	0.669	0.033
WineEW	0.989	0.000	0.960	0.012	0.951	0.015	0.919	0.052
Zoo	0.993	0.009	0.975	0.009	0.939	0.008	0.874	0.095
clean1	0.863	0.004	0.908	0.006	0.898	0.011	0.826	0.021
semeion	0.976	0.002	0.972	0.003	0.971	0.002	0.962	0.006
Colon	0.870	0.006	0.661	0.022	0.766	0.015	0.682	0.038
Leukemia	0.931	0.014	0.884	0.016	0.844	0.014	0.877	0.029

Table 11
Average number of selected attributes results of the BGOA_EPD_Tour and BGOA_EPD_RWS compared to other metaheuristics .

Dataset	BGOA_EPD_Tour		bGWO		BGSa		BBA	
	Att	StdDev	Att	StdDev	Att	StdDev	Att	StdDev
Breastcancer	5.000	0.000	7.100	1.447	6.067	1.143	3.667	1.373
BreastEW	17.333	2.440	19.000	4.307	16.567	2.979	12.400	2.762
Exactly	6.533	0.571	10.233	1.654	8.733	1.048	5.733	1.893
Exactly2	1.533	0.507	7.333	4.155	5.100	2.107	6.067	2.333
HeartEW	8.400	1.037	8.167	2.001	6.833	1.315	5.900	1.647
Lymphography	10.633	1.217	11.100	1.971	9.167	1.895	7.800	2.203
M-of-n	6.467	0.507	9.633	0.964	8.467	1.432	6.167	2.086
penglungEW	178.333	15.486	166.333	28.232	157.167	7.729	126.167	15.601
SonarEW	36.767	4.240	36.233	8.613	30.033	3.700	24.700	5.377
SpectEW	11.100	3.044	12.633	2.442	9.533	2.300	7.967	2.282
CongressEW	5.767	2.012	7.300	2.136	6.767	2.402	6.233	2.063
IonosphereEW	16.400	3.701	19.233	5.015	15.400	2.513	13.400	2.594
KrvskpEW	21.667	2.496	27.367	3.388	19.967	2.125	15.000	2.853
Tic-tac-toe	5.000	0.000	6.700	1.343	5.867	1.137	4.700	1.489
Vote	5.433	1.223	7.400	2.222	8.167	1.821	6.133	2.177
WaveformEW	26.233	3.451	31.967	4.612	19.900	2.917	16.667	3.304
WineEW	8.800	1.472	8.600	1.754	7.367	1.098	6.067	1.741
Zoo	9.167	1.967	10.367	2.484	8.167	1.177	6.567	2.501
clean1	92.600	7.802	121.267	20.691	83.700	5.421	64.767	10.016
semeion	157.033	11.485	200.100	31.022	133.533	7.422	107.033	10.947
Colon	1063.667	64.618	1042.100	126.721	995.833	20.021	827.500	55.371
Leukemia	3768.800	224.842	3663.767	294.872	3555.133	39.713	2860.000	247.642

erator assists the algorithm to maintain the diversity of grasshoppers. It also utilizes top solutions to guide the poor ones and this strategy improves the exploitative behavior of algorithm. Hence, the BGOA_EPD_Tour an enriched potential to retain and recover a stable balance between the exploration and exploitation phases in dealing with difficult feature spaces. The effect of selection operators are seen in the background of the improved results.

The proposed BGOA_EPD_Tour has also attained an acceptable StdDev values. Regarding the fitness, the bGWO show a good efficacy on 2 datasets. The penglungEW, Colon, Tic-tac-toe, krskpEW cases can be considered as relatively large datasets and the fitness values of the developed EPD-based version is relatively less than other optimizers.

Based on Table 13, the BGSa is the fastest approach and the binary BBA is placed at the next rank.

Table 14 shows all datasets that the proposed method provides the best results (20 cases), the improvements are meaningful and accuracy of classification results has significantly increased compared to the other competitors.

The convergence curves for the compared algorithms on all datasets are demonstrated in Figs. 11 and 12. It can be seen that the BGOA_EPD_Tour has an accelerated behavior on all 22 problems. Based on the last found solutions, which can be seen from the ending points of the curves, it can outperform all techniques in tackling 20 problems except the BreastEW and Clean1 datasets, which no stagnation behavior occurs but the concluding marks are not better than the bGWO algorithm. Premature convergence behaviors can be detected in the curves of the bGWO, BBA and BGSa in dealing with several cases including the Exactly, Vote, Tic-tac-toe, SpectEW, and Zoo cases. For 20 datasets, the curves BGOA_EPD_Tour are superior to those of other competitors. Based

Table 12

Average fitness results of the BGOA_EPD_Tour and BGOA_EPD_RWS compared to other metaheuristics.

Dataset	BGOA_EPD_Tour		bGWO		BGSA		BBA	
	Fitness	StdDev	Fitness	StdDev	Fitness	StdDev	Fitness	StdDev
Breastcancer	0.026	0.001	0.039	0.003	0.049	0.003	0.044	0.005
BreastEW	0.058	0.004	0.051	0.007	0.063	0.006	0.056	0.006
Exactly	0.006	0.006	0.197	0.077	0.307	0.059	0.323	0.074
Exactly2	0.219	0.000	0.260	0.019	0.295	0.024	0.326	0.017
HeartEW	0.171	0.004	0.213	0.017	0.226	0.021	0.208	0.015
Lymphography	0.137	0.011	0.191	0.028	0.222	0.022	0.226	0.024
M-of-n	0.005	0.000	0.112	0.041	0.170	0.063	0.171	0.056
penglungEW	0.078	0.012	0.154	0.013	0.085	0.000	0.168	0.017
SonarEW	0.094	0.008	0.169	0.016	0.116	0.015	0.110	0.021
SpectEW	0.177	0.010	0.194	0.014	0.220	0.024	0.172	0.012
CongressEW	0.039	0.005	0.056	0.011	0.053	0.008	0.064	0.015
IonosphereEW	0.105	0.007	0.120	0.009	0.122	0.010	0.108	0.012
KrvskpEW	0.038	0.003	0.073	0.015	0.097	0.047	0.117	0.047
Tic-tac-toe	0.196	0.000	0.251	0.032	0.251	0.024	0.257	0.024
Vote	0.037	0.003	0.060	0.010	0.073	0.011	0.071	0.013
WaveformEW	0.267	0.003	0.283	0.007	0.307	0.014	0.304	0.014
WineEW	0.018	0.001	0.047	0.012	0.054	0.015	0.036	0.013
Zoo	0.012	0.008	0.032	0.009	0.065	0.008	0.042	0.015
clean1	0.141	0.004	0.099	0.006	0.106	0.010	0.156	0.013
semeion	0.030	0.001	0.036	0.003	0.034	0.002	0.033	0.003
Colon	0.134	0.006	0.341	0.022	0.237	0.014	0.279	0.035
Leukemia	0.073	0.014	0.120	0.016	0.160	0.013	0.085	0.023

Table 13

Average CPU time (seconds) results of the BGOA_EPD_Tour and BGOA_EPD_RWS compared to other metaheuristics.

Dataset	BGOA_EPD_Tour		bGWO		BGSA		BBA	
	Time	StdDev	Time	StdDev	Time	StdDev	Time	StdDev
Breastcancer	4.146	0.191	3.879	0.236	3.461	0.194	3.456	0.184
BreastEW	5.404	0.241	4.602	0.234	3.748	0.173	3.862	0.196
Exactly	6.014	0.277	6.154	0.275	4.876	0.311	4.956	0.272
Exactly2	5.942	0.257	6.050	0.380	5.033	0.374	5.224	0.394
HeartEW	3.505	0.178	2.739	0.174	2.812	0.182	2.767	0.201
Lymphography	3.427	0.162	2.602	0.161	2.589	0.150	2.634	0.145
M-of-n	6.173	0.271	6.157	0.241	5.138	0.293	4.892	0.361
penglungEW	19.560	0.822	7.717	0.432	3.060	0.171	4.166	0.238
SonarEW	5.801	0.245	3.677	0.193	2.703	0.146	2.850	0.191
SpectEW	3.775	0.164	2.869	0.176	2.723	0.162	2.809	0.161
CongressEW	4.109	0.205	3.309	0.172	3.217	0.188	3.244	0.158
IonosphereEW	4.735	4.735	3.564	0.184	2.921	0.145	3.018	0.148
KrvskpEW	57.043	1.743	78.113	4.606	49.534	2.527	47.923	2.576
Tic-tac-toe	5.097	0.222	6.208	0.622	4.344	0.278	4.237	0.278
Vote	3.640	0.179	2.851	0.154	2.824	0.152	2.834	0.178
WaveformEW	152.167	4.812	213.955	13.607	125.804	6.770	119.990	8.169
WineEW	3.766	0.301	2.628	0.173	2.585	0.171	2.624	0.141
Zoo	3.614	0.240	2.623	0.136	2.547	0.158	2.777	0.192
clean1	17.220	0.608	13.683	0.647	7.397	0.288	7.589	0.467
semeion	123.254	2.147	169.538	9.191	90.601	2.212	82.776	5.579
Colon	112.236	4.364	36.694	1.998	5.154	0.242	12.176	0.774
Leukemia	394.688	15.418	130.878	6.366	16.441	0.683	39.379	2.418

on the aforementioned remarks, it can be recognized that the novel EPD-based operators have strengthened the overall tradeoff between the exploratory and exploitative steps. Consequently, it alleviates the immature convergence drawbacks of the BGOA in dealing with FS problems.

4.3. Comparison with other meta-heuristics in the literature

In this part, the classification rates of the proposed EPD-based approach are compared to some reported results from the past literature. Table 15 compares the average classification results of the BGOA_EPD_Tour with other algorithms obtained from the previous specialized works. The BGOA_EPD_Tour is compared to the reported classification results of GA and PSO from [97] and results of the bGWO1, bGWO2, GA, and PSO from [98]. Note that the first

versions of GA and PSO were executed with exact settings in the implementation of the authors in [97]. While the results of other four approaches (bGWO1, bGWO2, GA, and PSO) in the table were obtained by the authors with the same datasets in [98].

From the results in Table 15, it is evident that the attained classification rates by the developed EPD-based approach in this work are higher than other optimizers on 21 datasets and have a substantial superiority compared to those of GWO, GA, and PSO algorithms. The rates of the BGOA_EPD_Tour are better than the results of GA and PSO in [97] for all 22 datasets. In comparison with the revealed results in [98], the BGOA-based algorithm provides better classification rates than the PSO, GA, and bGWO1 on all 18 available datasets and it can provide better than the bGWO2 on 17 datasets. These results also affirm that the proposed EPD-based and selection operators in the BGOA_EPD_Tour

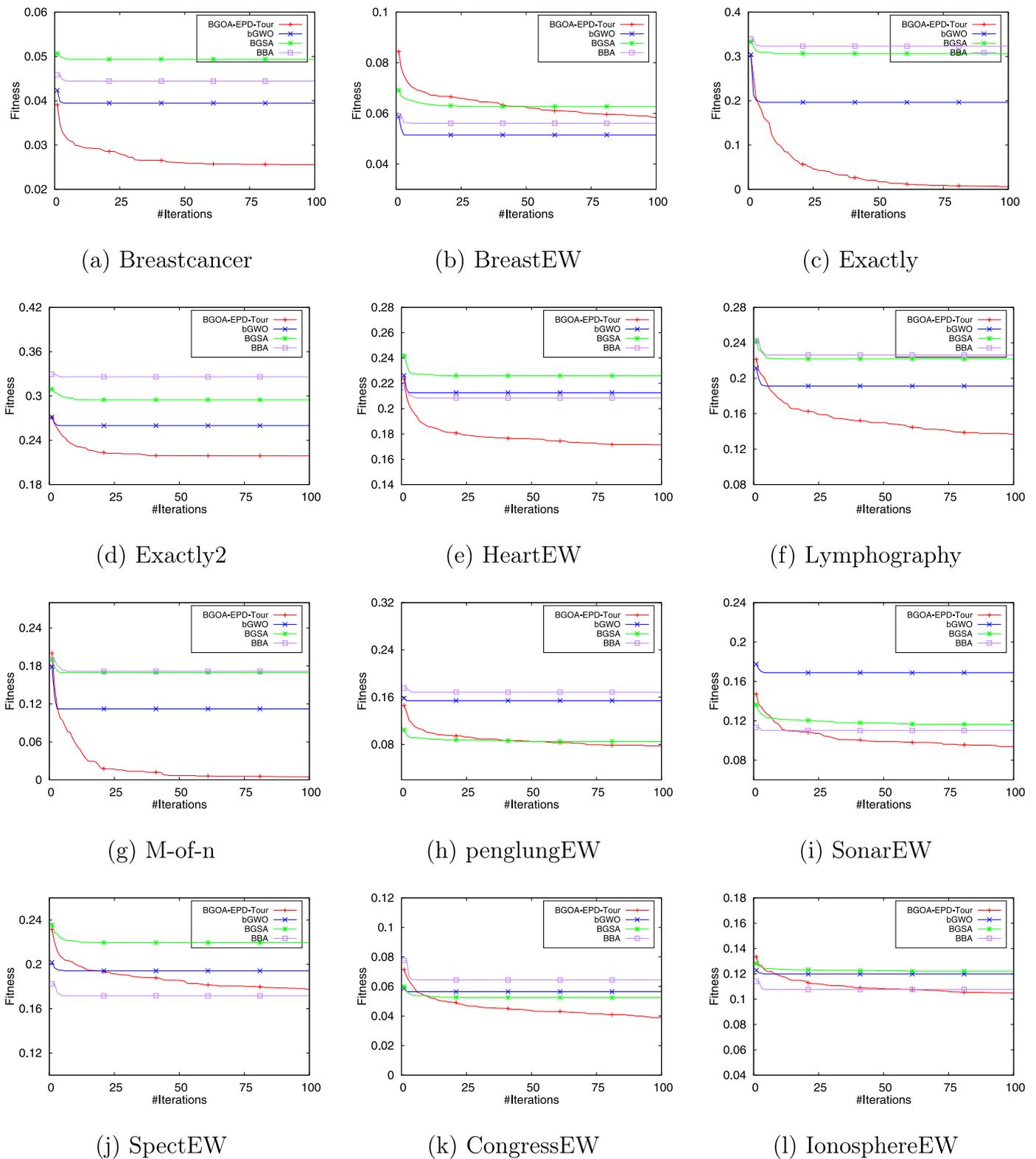


Fig. 11. Convergence curves for BGOA-EPD-Tour and other state-of-art methods for Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, and SonarEW, SpectEW, CongressEW, and IonosphereEW datasets.

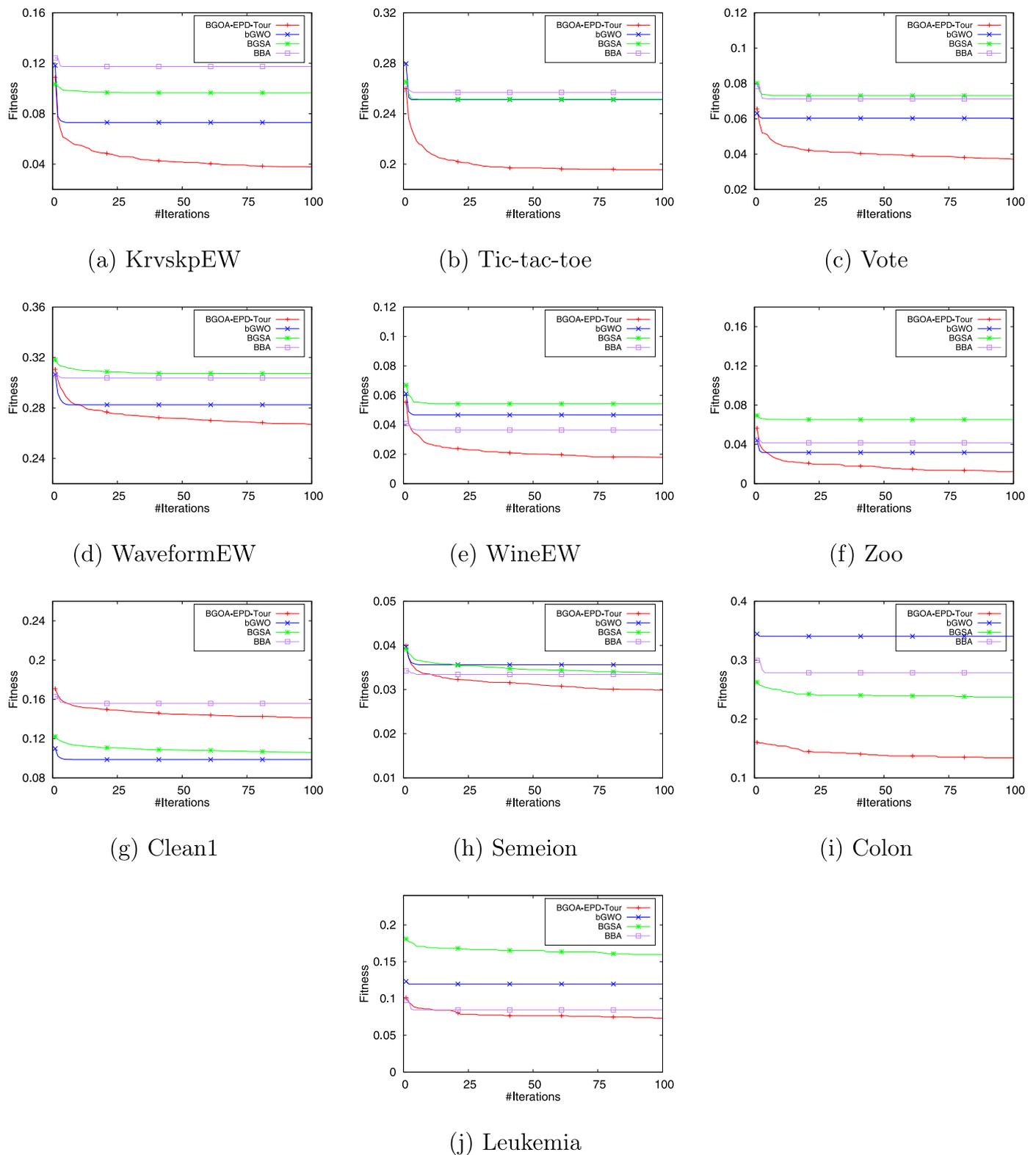


Fig. 12. Convergence curves for BGOA-EPD-Tour and other state-of-art methods for KrsvkpEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Zoo, Clean1, Semeion, Colon, and Leukemia datasets.

not only enriched its exploitation and exploitation capabilities and alleviated its stagnation problems but also enhanced the quality of the attained solutions for 22 datasets with various dimensions and characteristics.

4.4. Comparison with filter-based techniques

In this subsection, the classification of the EPD-embedded BGOA optimizer is compared to five well-known filter-based techniques [99]: correlation-based feature selection (CFS) [7], fast correlation-based filter (FCBF) [10], fisher score (F-score) [100], IG [101], and

Table 14

P-values of the Wilcoxon test the classification accuracy results of BGOA_EPD_Tour and other meta heuristics algorithms ($p \geq 0.05$ are underlined, and N/A means not applicable).

	BGOA_EPD_Tour	bGWO	BGSA	BBA
Breast Cancer	N/A	9.46E-13	1.40E-12	1.69E-12
BreastEW	1.04E-05	N/A	2.89E-08	1.46E-09
Exactly	N/A	2.13E-12	1.70E-12	1.71E-12
Exactly2	N/A	1.02E-12	1.18E-12	1.21E-12
HeartEW	N/A	2.77E-11	1.27E-10	1.44E-11
Lymphography	N/A	6.90E-10	1.89E-11	2.11E-11
M-of-n	N/A	1.20E-12	4.54E-12	1.21E-12
penglungEW	N/A	4.38E-12	1.31E-03	7.45E-12
SonarEW	N/A	1.74E-11	1.38E-08	8.57E-10
SpectEW	N/A	3.51E-06	4.25E-09	1.73E-05
CongressEW	N/A	1.32E-08	3.56E-08	2.30E-11
IonosphereEW	N/A	2.50E-07	1.65E-08	3.82E-07
KrvskpEW	N/A	3.69E-11	4.21E-10	2.88E-11
Tic-tac-toe	N/A	1.13E-12	1.17E-12	1.21E-12
Vote	N/A	4.12E-11	6.44E-12	7.66E-12
WaveformEW	N/A	1.06E-09	2.98E-11	2.99E-11
WineEW	N/A	7.91E-13	9.49E-13	4.39E-12
Zoo	N/A	1.93E-08	1.66E-12	1.71E-11
Clean1	2.05E-11	N/A	3.08E-05	2.61E-11
semeion	N/A	2.00E-07	1.99E-09	2.77E-10
Colon	N/A	8.62E-13	3.60E-13	1.25E-12
Leukemia	N/A	4.43E-11	5.63E-12	7.69E-10

wavelet power spectrum (Spectrum) [11]. These filter-based techniques have carefully selected from two main classes: univariate and multivariate approaches. The IG, Spectrum, and F-Score are from the univariate strategies, which do not reflect the dependencies of the features in the assessment measure. In addition, CFS and FCBF are from the other category, which can employ the dependencies of the features. These approaches are investigated here because they have different mechanisms for utilizing the class labels of the training info to realize the relevance of analyzed features. The supervised approaches such as CFS, FCBF, F-Score and IG can utilize class labels whereas the unsupervised techniques such as Spectrum cannot handle labels for assessing the features. The results for the filter-based techniques after 20 runs are compared with the rates of the BGOA_EPD_Tour in Table 16.

Inspecting the comparative results in Table 16, it is seen that the EPD-based approach can outperform other algorithms on 17

datasets, while the IG and F-Score methods have obtained the best results for 2 datasets. It surpassed the supervised univariate approaches such as F-Score and IG, and the supervised multivariate types such as CFS and FCBF, and the unsupervised Spectrum technique. Furthermore, the results indicate that the wrapper-based FS procedures can provide superior rates in comparison with the filter-based versions since they can utilize both labels and dependencies during the selection of associated subsets. It can be concluded that the proposed algorithm has merits among other well-regarded optimizers and outperforms some well-known filter-based approaches.

Taken together, the results and discussions showed that the binary operators integrated into the BGOA algorithm were beneficial. The proposed operators slightly change the solutions that accelerates local search and convergence of the proposed algorithm. The EPD operator randomly changes the worst solutions, which promotes diversity and global search of the proposed algorithm. In feature selection problems, the shape of search space changes for every new dataset. Feature selection is normally considered for problems with medium or large number of features as well. To handle these difficulties, therefore, we need an efficient optimization algorithm that shows less local optima stagnation and high accuracy. Both operators proposed in the method assist BGOA_EPD in handling these difficulties.

5. Conclusion and future directions

In this study, an efficient GOA-based optimizer with EPD and selection operators was proposed to improve the efficacy of the basic GOA in dealing with FS tasks. The proposed GOA_EPD approaches were utilized extensively to tackle 22 benchmark datasets. The overall classification accuracy, selected features, fitness, consumed CPU time, and convergence behaviors of all hybrid versions were compared in detail to select the best version of the BGOA_EPD. The BGOA_EPD_Tour and BGOA_EPD_RWS has obtained the best place among four developed hybrid variants. The BGOA_EPD_Tour technique was utilized and compared in detail to various well-known metaheuristic-based and filter-based FS methods. The comprehensive comparative results and analysis revealed the improved efficacy of the proposed algorithm for solving different FS tasks.

Table 15

Classification accuracies of the BGOA_EPD_Tour versus other meta-heuristics from the specialized literature.

	BGOA_EPD_Tour	GA [97]	PSO [97]	bGWO1[98]	bGWO2 [98]	GA [98]	PSO [98]
Breastcancer	0.980	0.957	0.949	0.976	0.975	0.968	0.967
BreastEW	0.947	0.923	0.933	0.924	0.935	0.939	0.933
Exactly	0.999	0.822	0.973	0.708	0.776	0.674	0.688
Exactly2	0.780	0.677	0.666	0.745	0.750	0.746	0.730
HeartEW	0.833	0.732	0.745	0.776	0.776	0.780	0.787
Lymphography	0.868	0.758	0.759	0.744	0.700	0.696	0.744
M-of-n	1.000	0.916	0.996	0.908	0.963	0.861	0.921
penglungEW	0.927	0.672	0.879	0.600	0.584	0.584	0.584
SonarEW	0.912	0.833	0.804	0.731	0.729	0.754	0.737
SpectEW	0.826	0.756	0.738	0.820	0.822	0.793	0.822
CongressEW	0.964	0.898	0.937	0.935	0.938	0.932	0.928
IonosphereEW	0.899	0.863	0.876	0.807	0.834	0.814	0.819
KrvskpEW	0.968	0.940	0.949	0.944	0.956	0.920	0.941
Tic-tac-toe	0.808	0.764	0.750	0.728	0.727	0.719	0.735
Vote	0.966	0.808	0.888	0.912	0.920	0.904	0.904
WaveformEW	0.737	0.712	0.732	0.786	0.789	0.773	0.762
WineEW	0.989	0.947	0.937	0.930	0.920	0.937	0.933
Zoo	0.993	0.946	0.963	0.879	0.879	0.855	0.861
clean1	0.863	0.862	0.845	-	-	-	-
semeion	0.976	0.963	0.967	-	-	-	-
Colon	0.870	0.682	0.624	-	-	-	-
Leukemia	0.931	0.705	0.862	-	-	-	-

- The results of this dataset are not available.

Table 16

Classification accuracy results of all filter-based methods versus the BGOA_EPD_Tour algorithm.

Dataset	CFS	FCBF	F-Score	IG	Spectrum	BGOA_EPD_Tour
Breastcancer	0.957	0.986	0.979	0.957	0.957	0.980
BreastEW	0.825	0.798	0.930	0.930	0.772	0.947
Exactly	0.670	0.440	0.600	0.615	0.575	0.999
Exactly2	0.705	0.545	0.680	0.620	0.660	0.780
HeartEW	0.648	0.648	0.759	0.759	0.796	0.833
Lymphography	0.500	0.567	0.667	0.667	0.767	0.868
M-of-n	0.785	0.815	0.815	0.815	0.580	1.000
PenglungEW	0.600	0.667	0.800	0.667	0.400	0.927
SonarEW	0.310	0.214	0.048	0.191	0.048	0.912
SpectEW	0.736	0.774	0.793	0.793	0.736	0.826
CongressEW	0.793	0.793	0.908	0.828	0.828	0.964
IonosphereEW	0.857	0.857	0.729	0.800	0.829	0.899
KrvskpEW	0.768	0.934	0.959	0.934	0.377	0.968
Tic-tac-toe	0.000	0.000	0.010	0.010	0.167	0.808
Vote	0.950	0.950	0.933	0.967	0.850	0.966
WaveformEW	0.620	0.710	0.662	0.662	0.292	0.737
WineEW	0.778	0.889	0.861	0.889	0.889	0.989
Zoo	0.800	0.900	0.650	0.850	0.600	0.993
clean1	0.716	0.642	0.632	0.547	0.611	0.863
semeion	0.875	0.875	0.875	0.868	0.875	0.976
Colon	0.750	0.667	0.667	0.667	0.500	0.870
Leukemia	0.929	0.857	0.980	0.980	0.357	0.931

Future studies can focus on the application of the EPD strategy to other population-based optimizers. The efficacy of the proposed binary GOA and EPD-based algorithms can also be employed to tackle other data mining problems. For future works, we intended to compare the proposed GOA-EPD with different classes of FS methods in the field.

References

- [1] Y. Huang, T. Li, C. Luo, H. Fujita, S.-j. Horng, Dynamic variable precision rough set approach for probabilistic set-valued information systems, *Knowl. Based Syst.* 122 (2017) 131–147.
- [2] L. Jiao, F. Shang, F. Wang, Y. Liu, Fast semi-supervised clustering with enhanced spectral embedding, *Pattern Recognit.* 45 (2012) 4358–4369.
- [3] J. Han, J. Pei, M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.
- [4] H. Wang, Z. Xu, W. Pedrycz, An overview on the roles of fuzzy set techniques in big data processing: trends, challenges and opportunities, *Knowl. Based Syst.* 118 (2017) 15–30.
- [5] H. Liu, H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, Boston, 1998.
- [6] Y. Zhang, S. Wang, P. Phillips, G. Ji, Binary pso with mutation operator for feature selection using decision tree applied to spam detection, *Knowl. Based Syst.* 64 (2014) 22–31.
- [7] M.A. Hall, L.A. Smith, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, in: *FLAIRS conference*, volume 1999, 1999, pp. 235–239.
- [8] R. Shang, W. Wang, R. Stolkin, L. Jiao, Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection, *IEEE Trans. Cybern.* (2017).
- [9] F. Bellal, H. Elghazel, A. Aussem, A semi-supervised feature ranking method with ensemble learning, *Pattern Recognit. Lett.* 33 (2012) 1426–1433.
- [10] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 856–863.
- [11] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, *Proceedings of the 24th International Conference on Machine Learning, ACM*, 2007, pp. 1151–1157.
- [12] R. Shang, W. Wang, R. Stolkin, L. Jiao, Subspace learning-based graph regularized feature selection, *Knowl. Based Syst.* 112 (2016) 152–165.
- [13] R. Shang, Z. Zhang, L. Jiao, C. Liu, Y. Li, Self-representation based dual-graph regularized feature selection clustering, *Neurocomputing* 171 (2016) 1242–1253.
- [14] J. Ren, Z. Qiu, W. Fan, H. Cheng, S.Y. Philip, Forward semi-supervised feature selection, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2008, pp. 970–976.
- [15] M. Dash, H. Liu, Feature selection for classification, *Intell. Data Anal.* 1 (1997) 131–156.
- [16] V. Bolón-Canedo, N. Sánchez-Marroño, A. Alonso-Betanzos, Recent advances and emerging challenges of feature selection in the context of big data, *Knowl. Based Syst.* 86 (2015) 33–45.
- [17] H. Liu, R. Setiono, Chi2: feature selection and discretization of numeric attributes, 1995.
- [18] J. Quinlan, Induction of decision trees, *Mach. Learn.* 1 (1986) 81–106.
- [19] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
- [20] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relief and rrelieff, *Mach. Learn.* 53 (2003) 23–69.
- [21] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Accelerating wrapper-based feature selection with k-nearest-neighbor, *Knowl. Based Syst.* 83 (2015) 81–91.
- [22] H. Liu, R. Setiono, Feature selection and classification—a probabilistic wrapper approach, 1996.
- [23] R. Setiono, H. Liu, Neural-network feature selector, *Neural Netw. IEEE Trans.* 8 (1997) 654–662.
- [24] W. Siedlecki, J. Sklansky, On automatic feature selection, *Int. J. Pattern Recognit. Artif. Intell.* 2 (1988) 197–220.
- [25] P. Langley, Selection of relevant features in machine learning, 1994.
- [26] C. Lai, M.J. Reinders, L. Wessels, Random subspace method for multivariate feature selection, *Pattern Recognit. Lett.* 27 (2006) 1067–1076.
- [27] E. Talbi, *Metaheuristics from Design to Implementation*, Wiley Online Library, 2009.
- [28] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, *Neural Comput. Appl.* 26 (2015) 1257–1263.
- [29] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [30] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [31] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Softw.* 83 (2015) 80–98.
- [32] X.-S. Yang, *Firefly Algorithms for Multimodal Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 169–178.
- [33] J. Kennedy, R. Eberhart, A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, MHS '95, 1995, pp. 39–43.
- [34] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Mag.* 1 (2006) 28–39.
- [35] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* 3 (2003) 1157–1182.
- [36] E. Zorarpaci, S.A. Özel, A hybrid approach of differential evolution and artificial bee colony for feature selection, *Expert Syst. Appl.* 62 (2016) 91–103.
- [37] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, H. Yang, Path planning for solar-powered UAV in urban environment, *Neurocomputing* (2017).
- [38] W. Hang, K.-S. Choi, S. Wang, Synchronization clustering based on central force optimization and its extension for large-scale datasets, *Knowl. Based Syst.* 118 (2017) 31–44.
- [39] I. Aljarah, S.A. Ludwig, A new clustering approach based on glowworm swarm optimization, in: *2013 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2013, pp. 2642–2649.
- [40] I. Aljarah, S.A. Ludwig, Parallel particle swarm optimization clustering algorithm based on mapreduce methodology, 2012 fourth world congress on Nature and biologically inspired computing (NaBiC), IEEE, 2012, pp. 104–111.
- [41] H. Faris, I. Aljarah, S. Mirjalili, Improved monarch butterfly optimization for unconstrained global search and neural network training, *Appl. Intell.* (2017) 1–20.
- [42] H. Faris, M.A. Hassonah, A.-Z. Ala'M, S. Mirjalili, I. Aljarah, A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture, *Neural Comput. Appl.* 1–15.

- [43] H. Faris, I. Aljarah, S. Mirjalili, Training feedforward neural networks using multi-verse optimizer for binary classification problems, *Appl. Intell.* 45 (2) (2016) 322–332.
- [44] I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Comput.* (2016) 1–15.
- [45] Y.Z. Li, L. Jiang, Q. Wu, P. Wang, H.B. Gooi, K. Li, Y. Liu, P. Lu, M. Cao, J. Imura, Wind-thermal power system dispatch using mlsad model and gsoiclw algorithm, *Knowl. Based Syst.* 116 (2017) 94–101.
- [46] E. Ali, S.A. Elazim, A. Abdelaziz, Ant lion optimization algorithm for optimal location and sizing of renewable distributed generations, *Renew. Energy* 101 (2017) 1311–1324.
- [47] H. Faris, I. Aljarah, M.A. Al-Betar, S. Mirjalili, Grey wolf optimizer: a review of recent variants and applications, *Neural Comput. Appl.* (2017).
- [48] K. Sorensen, M. Sevaux, F. Glover, A history of metaheuristics, 2017, *ArXiv preprint ArXiv:1704.00853*.
- [49] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Softw.* 105 (2017) 30–47.
- [50] A.A. El-Fergany, Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser, *IET Renewable Power Gener.* (2017).
- [51] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, Y. Yu, Distributed trajectory optimization for multiple solar-powered uavs target tracking in urban environment by adaptive grasshopper optimization algorithm, *Aerosp. Sci. Technol.* 70 (2017) 497–510.
- [52] A. Tharwat, E.H. Houssein, M.M. Ahmed, A.E. Hassanien, T. Gabel, Mogo algorithm for constrained and unconstrained multi-objective optimization problems, *Appl. Intell.* (2017) 1–16.
- [53] S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, *Appl. Intell.* (2017) 1–16.
- [54] S. Mirjalili, Sca: a sine cosine algorithm for solving optimization problems, *Knowl. Based Syst.* 96 (2016) 120–133.
- [55] H. Faris, I. Aljarah, N. Al-Madi, S. Mirjalili, Optimizing the learning process of feedforward neural networks using lightning search algorithm, *Int. J. Artif. Intell. Tools* 25 (6) (2016) 1650033.
- [56] T. Bäck, D. Fogel, Z. Michalewicz, *Handbook of Evolutionary Computation*, Release 97 (1997) B1.
- [57] A.-C. Zăvoianu, E. Lughofer, W. Koppelstätter, G. Weidenholzer, W. Amrhein, E.P. Klement, Performance comparison of generational and steady-state asynchronous multi-objective evolutionary algorithms for computationally-intensive problems, *Knowl. Based Syst.* 87 (2015) 47–60.
- [58] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.
- [59] L.J. Fogel, A.J. Owens, M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, Wiley, 1966.
- [60] C.-F. Tsai, W. Eberle, C.-Y. Chu, Genetic algorithms in feature and instance selection, *Knowl. Based Syst.* 39 (2013) 240–247.
- [61] P. Moradi, M. Rostami, Integration of graph clustering with ant colony optimization for feature selection, *Knowl. Based Syst.* 84 (2015) 144–161.
- [62] H. Wang, X. Jing, B. Niu, A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data, *Knowl. Based Syst.* 126 (2017) 8–19.
- [63] R. Jensen, Q. Shen, Finding rough set reducts with ant colony optimization, in: *Proceedings of the 2003 UK workshop on computational intelligence*, volume 1, pp. 15–22, 2003.
- [64] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, *IEEE Trans. Knowl. Data Eng.* 16 (2004) 1457–1471.
- [65] H. Chen, W. Jiang, C. Li, R. Li, A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm, *Math. Problems Eng.* 2013 (2013).
- [66] M. Mafarja, S. Abdullah, Investigating memetic algorithm in solving rough set attribute reduction, *Int. J. Comput. Appl. Technol.* 48 (2013) 195–202.
- [67] M. Majdi, S. Abdullah, N.S. Jaddi, Fuzzy population-based meta-heuristic approaches for attribute reduction in rough set theory, *World Academy of Science, Engineering and Technology, Int. J. Comput. Electr. Autom. Control Inf. Eng.* 9 (2015) 2289–2297.
- [68] J. Wang, A.-R. Hedar, S. Wang, Scatter search for rough set attribute reduction, *IEEE*, in: *Bio-Inspired Computing: Theories and Applications*, 2007. BIC-TA, 2007, pp. 236–240. Second International Conference on
- [69] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Softw.* 83 (2015) 80–98.
- [70] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary antlion approaches for feature selection, *Neurocomputing* (2016).
- [71] H.M. Zawbaa, E. Emary, B. Parv, Feature selection based on antlion optimization algorithm, 2015.
- [72] H.M. Zawbaa, E. Emary, C. Grosan, Feature selection via chaotic antlion optimization, *PLoS ONE* 11 (2016) e0150652.
- [73] R.E. Precup, R.C. David, E.M. Petriu, Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity, *IEEE Trans. Ind. Electron.* 64 (2017) 527–534.
- [74] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing* 172 (2016) 371–381.
- [75] E. Emary, H.M. Zawbaa, C. Grosan, A.E. Hassanien, Feature subset selection approach by gray-wolf optimization, in: *Afro-European Conference for Industrial Advancement*, Springer pp. 1–13, 2015.
- [76] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, *Neurocomputing* (2017).
- [77] A. Lewis, S. Mostaghim, M. Randall, Evolutionary population dynamics and multi-objective optimisation problems, *Multi-Object. Optim. Comput. Intell.* (2008) 185–206.
- [78] S. Boettcher, A.G. Percus, Extremal optimization: methods derived from co-evolution, in: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, Morgan Kaufmann Publishers Inc., 1999, pp. 825–832.
- [79] M. Randall, T. Hendtlass, A. Lewis, Extremal optimisation for assignment type problems, *Biol.-Inspired Optim. Methods* 210 (2009) 139–164.
- [80] D. Xie, Z. Luo, F. Yu, The computing of the optimal power consumption for semi-track air-cushion vehicle using hybrid generalized extremal optimization, *Appl. Math. Model.* 33 (2009) 2831–2844.
- [81] K. Tamura, H. Kitakami, A. Nakada, Distributed modified extremal optimization using island model for reducing crossovers in reconciliation graph, *Eng. Lett.* 21 (2013).
- [82] P. Bak, C. Tang, K. Wiesenfeld, Self-organized criticality: an explanation of the 1/f noise, *Phys. Rev. Lett.* 59 (1987) 381.
- [83] A. Lewis, D. Abramson, T. Peachey, An evolutionary programming algorithm for automatic engineering design, in: *International Conference on Parallel Processing and Applied Mathematics*, Springer, 2003, pp. 586–594.
- [84] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, *Appl. Soft Comput.* 60 (2017) 115–134.
- [85] A.A. Heidari, R.A. Abbaspour, A.R. Jordehi, An efficient chaotic water cycle algorithm for optimization tasks, *Neural Comput. Appl.* 28 (2017) 57–85.
- [86] S.K. Pal, A. Skowron, *Rough-fuzzy Hybridization: A New Trend in Decision Making*, Springer-Verlag New York, Inc., 1999.
- [87] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, *Swarm Evol. Comput.* 9 (2013) 1–14.
- [88] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation*, 1997, pp. 4104–4108. IEEE International Conference on, volume 5, IEEE.
- [89] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *Am. Stat.* 46 (1992) 175–185.
- [90] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford university press, 1996.
- [91] B.L. Miller, D.E. Goldberg, et al., Genetic algorithms, tournament selection, and the effects of noise, *Complex Syst.* 9 (1995) 193–212.
- [92] A. Frank, A. Asuncion, UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California. *School of information and computer science* 213 (2010).
- [93] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, *Neurocomputing* 213 (2016) 54–65.
- [94] J. Friedman, T. Hastie, R. Tibshirani, *The elements of statistical learning*, Series in Statistics New York, 1, Springer, 2001.
- [95] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Bgsa: binary gravitational search algorithm, *Nat. Comput.* 9 (2010) 727–745.
- [96] S. Mirjalili, S.M. Mirjalili, X.-S. Yang, Binary bat algorithm, *Neural Comput. Appl.* 25 (2014) 663–681.
- [97] S. Kashaf, H. Nezamabadi-pour, An advanced aco algorithm for feature subset selection, *Neurocomputing* 147 (2015) 271–279.
- [98] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing* 172 (2016) 371–381.
- [99] Y. Saeyns, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, *Bioinformatics* 23 (2007) 2507–2517.
- [100] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley & Sons, 2012.
- [101] T.M. Cover, J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 2012.