Binary grasshopper optimisation algorithm approaches for feature selection problems

Majdi Mafarja [a], Ibrahim Aljarah [b], Hossam Faris [b], Abdelaziz I. Hammouri [c], Ala' M. Al-Zoubi [b], Seyedali Mirjalili [d],*

[a] *Department of Computer Science, Birzeit University, Birzeit, Palestine*
[b] *King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*
[c] *Department of Computer Information Systems, Al-Balqa Applied University, Al-Salt, Jordan*
[d] *Institute for Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia*

ARTICLE INFO

ABSTRACT

Feature Selection (FS) is a challenging machine learning-related task that aims at reducing the number of features by removing irrelevant, redundant and noisy data while maintaining an acceptable level of classification accuracy. FS can be considered as an optimisation problem. Due to the difficulty of this problem and having a large number of local solutions, stochastic optimisation algorithms are promising techniques to solve this problem. As a seminal attempt, binary variants of the recent Grasshopper Optimisation Algorithm (GOA) are proposed in this work and employed to select the optimal feature subset for classification purposes within a wrapper-based framework. Two mechanisms are employed to design a binary GOA, the first one is based on Sigmoid and V-shaped transfer functions, and will be indicated by BGOA-S and BGOA-V, respectively. While the second mechanism uses a novel technique that combines the best solution obtained so far. In addition, a mutation operator is employed to enhance the exploration phase in BGOA algorithm (BGOA-M). The proposed methods are evaluated using 25 standard UCI datasets and compared with 8 well-regarded metaheuristic wrapper-based approaches, and six well known filter-based (*e.g.,* correlation FS) approaches. The comparative results show the superior performance of the BGOA and BGOA-M methods compared to other similar techniques in the literature.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The existence of huge amounts of data available in real-world problems makes the process of analysing them challenging and often inaccurate; due to irrelevant or redundant data (Li, Li & Liu, 2017). Feature Selection (FS) is a preprocessing phase that eliminates the redundant and irrelevant features and finds a final set of the most informative features, leading to better performance in data mining techniques (*i.e.,* classification) (Han, Pei & Kamber, 2011).

FS for classification problems is a challenging and computationally expensive process, especially when dealing with high-dimensional datasets (Dash & Liu, 1997). This is because the classification accuracy obtained by a set of selected features must be higher than that obtained by the full set of the features. FS process aims at minimizing the number of features, which directly reduces the size of the search space/landscape and assists machine learning techniques to utilise the most important features only (Liu & Motoda, 1998).

The goal of FS is to find a subset of M features from the full set with N features, this improves the performance of the learning algorithm (learning speed and/or classification accuracy). FS can be considered as an optimisation problem; since it searches for the (near) optimal subset. Therefore, exhaustive (exact) search strategies are impractical in this area; as it generates all possible solutions to obtain only one best solution (Liu & Motoda, 1998). For instance, if a dataset contains N features, then 2^N solutions should be generated and evaluated, which requires high computational cost (Guyon & Elisseeff, 2003). Much work can be found in the literature trying to implement stochastic methods to tackle the FS problem; mainly, due to better local optima avoidance compared to the conventional optimisation techniques (Asir, Appavu & Jebamalar, 2016). The superiority of metaheuristics originates from

having the ability to find acceptable solutions in reasonable periods of time (Talbi, 2009).

Metaheuristics are stochastic algorithms and divided into two categories: population-based and single-solution-based algorithms (Glover & Kochenberger, 2006; Talbi, 2009). In population-based algorithms (*e.g.,* Genetic Algorithms, Swarm Intelligence techniques), a set of solutions called population is evolved, while one solution is manipulated and transformed in single-solution-based algorithms (*e.g.,* Simulated Annealing and Tabu Search) (Talbi, 2009). Various metaheuristics including Tabu Search (TS) (Hedar, Wang & Fukushima, 2008), Simulated Annealing (SA) (Jensen & Shen, 2004), Genetic Algorithm (GA) (Kabir, Shahjahan & Murase, 2011), Particle Swarm Optimisation (PSO) (Moradi & Gholampour, 2016), Ant Colony Optimisation (ACO) (Kashef & Nezamabadi-pour, 2015), Differential Evolution (DE) (Zorarpacı & Özel, 2016), and Artificial Bee Colony (ABC) (Wang, Li & Ren, 2010) have all been used in the literature to search in feature subset space for selecting the (sub)optimal feature set.

Searching for the best feature set requires an evaluation criterion to compare the subsets and find the best one. From evaluation point of view, FS can be divided into two categories: filters and wrappers (Liu & Motoda, 1998). Filter-based methods evaluate feature subsets independently from a learning algorithm and depend on the data itself using designated methods (*e.g.,* F-score, Information Gain (IG) and Principal Component Analysis (PCA) (Katrutsa & Strijov, 2017; Song, Jiang & Liu, 2017). In contrast, in wrapper methods, a learning algorithm (*e.g.,* classification) is utilised to evaluate the feature subset during the search process (Kohavi & John, 1997). Filters are faster than wrappers because they do not require extra time to call the learning algorithm (Liu & Motoda, 1998). However, wrappers have been widely investigated for classification accuracy, since they are proved to be beneficial in finding feature subsets that suite a predetermined classifier (Liu & Yu, 2005). Generally speaking, three factors should be determined when using a wrapper FS model: a classifier (*k*-Nearest Neighbors, Support Vector Machine, Decision Tree, etc.), feature subset evaluation criteria (accuracy, false-positive elimination rate, area under the ROC), and a searching (optimisation) technique to find the best combination of features (Zarshenas & Suzuki, 2016).

Another successful algorithm in both optimisation and FS fields is Moth-flame Optimisation (MFO) algorithm (Zawbaa, Emary & Grosan, 2016). ALO, GWO, and MFO are examples of nature-inspired algorithms that mimic the behaviour of these species in seeking their food sources (optimal solutions). Whale Optimization Algorithm (WOA)-based FS approaches have been proposed in Mafarja and Mirjalili (2017). In this paper, different hybridisation models between WOA and Simulated Annealing algorithm were proposed and used for FS problem.

Genetic Algorithm (GA) (Holland, 1992) is considered as the first evolution-based algorithm that is capable of solving non-linear and complex problems (Yang & Honavar, 1998). Different binary GA approaches have been proposed to tackle the FS in the literature (Oh, Lee & Moon, 2004; Yang & Honavar, 1998). Also, a Chaos Genetic Feature Selection Optimisation method (CGFSO) is presented in Chen, Jiang, Li and Li (2013). Besides GA, Particle Swarm Optimisation (PSO) algorithm, proposed by Kennedy (1995), was designed to tackle the continuous optimisation problems. A binary version of PSO (BPSO) was proposed in Kennedy and Eberhart (1997) to deal with binary optimisation problems. BPSO was used in FS approaches as in Chakraborty (2008); Wang, Yang, Teng, Xia and Jensen (2007); Xue, Zhang and Browne (2014). Moreover, Aghdam, Ghasem-Aghaee and Basiri (2009) proposed a FS approach based on Ant Colony Optimization (ACO) algorithm. Using benchmark datasets, their results showed the superiority of the ACO algorithm. Moreover, Kabir, Shahjahan and Murase (2009) proposed an efficient FS approach using ACO algorithm called ACOFS

where a new pheromone update mechanism has been used The results proved the superiority of this approach. A similar approach by the same authors was proposed in Kabir, Shahjahan and Murase (2012). For more ACO approaches for FS problem, readers can refer to Dadaneh, Markid and Zakerolhosseini (2016); Wan, Wang, Ye and Lai (2016).

Based on the biological behaviour of bees, Karaboga (2005) proposed an optimisation approach called the Artificial Bee Colony (ABC) algorithm. Additionally, Shokouhifar and Sabet (2010) described a hybrid approach for efficient FS using neural networks and ABC optimisation. A FS for intrusion detection systems using binary ABC is also proposed in Wang et al. (2010). Furthermore, A binary version of the Gravitational Search Algorithm (BGSA) was proposed to tackle FS (Rashedi, Nezamabadi-Pour & Saryazdi, 2010). While two wrapper FS approaches based on Scatter Search Algorithm (SSA) were introduced in López, Torres, Batista, Pérez and Moreno-Vega (2006) and López, Torres, Pérez and Vega (2004). Another FS approach based on SSA was proposed by Wang, Hedar, Wang and Ma (2012).

The research area of adapting and applying metaheuristics to FS problems is ongoing, and many newly proposed approaches can be found in the literature. Recently, Lu, Chen, Yan, Jin, Xue and Gao (2017) proposed a novel hybrid FS algorithm that uses Mutual information and GA. Another FS approach based on Cohort Intelligence (CI) Algorithm (Kulkarni, Durugkar & Kumar, 2013) was proposed in Aladeemy, Tutun and Khasawneh (2017). Moreover, a hybrid approach algorithm combining artificial bee colony algorithm and differential evolution algorithm was proposed as a wrapper FS method in Zorarpacı and Özel (2016). More recent FS approaches can be found in these papers Baig, Aslam, Shum and Zhang (2017); Ghareb, Bakar and Hamdan (2016); Wang, Ci and YAO (2005). Here, one might ask if there is still an opportunity to develop a new metaheuristic-based algorithm for FS problems. According to No-Free-Lunch (NFL) theorem (Wolpert & Macready, 1997), there is *no algorithm* for solving *all optimisation problems*. This statement means that a FS developed using a metaheuristic might show poor performance when the datasets are changed. Therefore, there is always an opportunity to propose new metaheuristic-based FS methods to enhance the process of solving FS problems. This motivated our attempts to propose a new FS method based on the recent Grasshopper Optimisation Algorithm (GOA) as presented in the following sections.

Grasshopper Optimization Algorithm is a recent nature-inspired algorithm that mimics the swarming behaviour of grasshoppers in nature (Saremi, Mirjalili & Lewis, 2017). The original version of GOA was designed for solving continuous optimisation problems (Saremi et al., 2017). However, many optimisation problems (*e.g.,* FS) have discrete decision variables and search space. In this paper, new binary versions of GOA are proposed and designed to solve the FS problem. In the first two approaches, two transfer functions that belong to two families (*i.e.,* $S-shaped$ and $V-shaped$) are used to convert the continuous solutions in GOA to binary ones. These two approaches are called BGOA-S and BGOA-V. A novel mechanism is used with the third approach; the binary version of the solution is generated with respect to the location of the best solution found so far (target) and the step vector, instead of depending on the step vector as in the first two approaches. This approach is called BGOA. Moreover, to enhance the exploratory behaviour of the GOA algorithm, a mutation operator is employed with the BGOA in a fourth approach called BGOA-M. The performance of the proposed approaches is evaluated and compared with similar methods in the literature, and with some thoughtfully selected standard FS approaches.

The rest of this paper is organised as follows: Section 2 presents the basics of the GOA algorithm. The proposed binary versions of GOA are discussed in Section 3. In Section 4, the experimen-

tal results are presented, and the results are analysed. Finally, in Section 5, conclusions and future work are given.

2. Grasshopper optimisation algorithm (GOA)

Grasshoppers are insects that are considered as plant-eaters. Grasshoppers are usually seen individually in nature, however, when they join in one swarm, which may consist of millions of grasshoppers, they become a severe pest; due to their damage to crops, pasture and grain (Simpson, McCAFFERY & HAeGELE, 1999). The life cycle of a grasshopper consists of three stages; egg, nymph, and adult. The grasshopper can be found in the swarm in nymph or adulthood life phases (Rogers, Matheson, Despland, Dodgson, Burrows & Simpson, 2003). The swarm in the nymph phase is characterised by slow movement with small steps by the grasshoppers. By contrast, the swarm in the adult phase is characterised by abrupt and long distance movement. Seeking food sources is a critical characteristic of the swarming grasshoppers.

Exploration and exploitation are the two main phases in the nature-inspired algorithms that aim to improve the convergence speed and/or local optima avoidance in the algorithm when searching for a target. In the exploitation process, the search agents tend to move locally in the search space. In contrast, they are encouraged to move abruptly during the exploration process. Grasshoppers perform these two processes as well as naturally seeking the target (food source). Eq. (1) presents the simulation of the swarming behaviour of grasshoppers (Saremi et al., 2017).

$$X_i = r_1 S_i + r_2 G_i + r_3 A_i \tag{1}$$

where X_i defines the position of the $i-th$ grasshopper, S_i is the social interaction defined in Eq. (2), G_i is the gravity force on the $i-th$ grasshopper in Eq. (4), A_i shows the wind advection Eq. (5), and r_1, r_2, and r_3 are random numbers in the interval [0,1].

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

where d_{ij} is the Euclidean distance between the $i-th$ and the $j-th$ grasshopper that 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. s is defined as the strength of social forces calculated as in Eq. (3):

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

where f indicates the intensity of attraction, and l is the attractive length scale. More details about the impact of these two parameters (*i.e.*, attraction and repulsion) on the social behaviours of artificial grasshoppers can be found in the original GOA paper (Saremi et al., 2017). The authors presented extensive experiments to study the behaviour of the grasshoppers with different values of l and f, and they found that the repulsion occurs between any two grasshoppers if the distance between them is in the interval [0,2.079]. If a grasshopper is 2.079 units away from other grasshoppers, it enters the comfort zone.

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

where g is the gravitational constant and $\widehat{e_g}$ shows a unity vector towards the centre of the earth.

$$A_i = u \times \widehat{e_w} \tag{5}$$

where u is a constant drift and $\widehat{e_w}$ is a unit vector in the direction of the wind. Nymph grasshoppers have no wings, so their movements are highly correlated with wind direction.

Fig. 1 illustrates the conceptual model of the comfort zone and the attraction and repulsion forces between grasshoppers. Eq. (6) presents how to determine the new position of the $i-th$
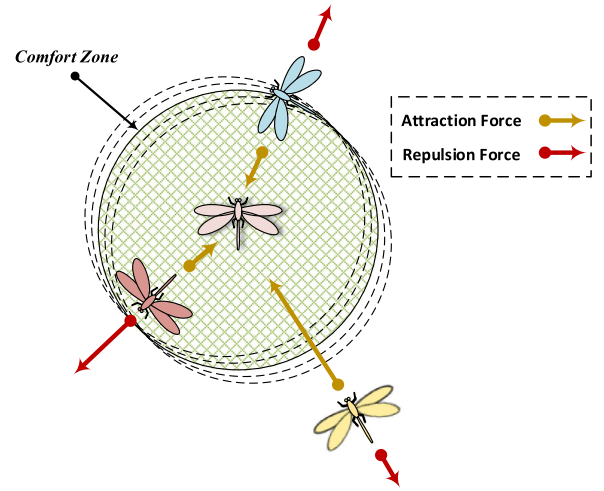

Fig. 1. Primitive corrective patterns between individuals in a swarm of grasshoppers.

grasshopper based on its current position, the position of all other grasshoppers, and the position of the target (food source).

$$X_i^d = c_1 \left(\sum_{j=1,\, j\neq i}^{N} c_2 \frac{ub_d - lb_d}{2} s\left(|x_j^d - x_i^d|\right) \frac{x_j - x_i}{d_{ij}} \right) + \widehat{T_d} \tag{6}$$

where ub_d is the upper bound in the Dth dimension, lb_d is the lower bound in the Dth dimension, $s(d) = f e^{\frac{-d}{l}} - e^{-d}$; $\widehat{T_d}$ is the value of the Dth dimension in the target (best solution found so far), and c is a decreasing coefficient to shrink the comfort zone, attraction region and repulsion region. Note that S is similar to the S component in Eq. (1). However, the gravity component G has not been considered, and the wind direction A has been assumed towards the target $\widehat{T_d}$.

In Eq. (6), the adaptive parameter c has been used twice to simulate the deceleration of grasshoppers approaching the source of food and eventually consuming it. The outer c_1 is used to reduce the search coverage toward the target grasshopper as the iteration count increases, while the inner c_2 has been used to reduce the effect of the attraction and repulsion forces between grasshoppers proportionally to the number of iterations.

The mathematical formulations of GOA are capable of efficiently exploiting and exploring the search space. However, it still needs a mechanism for tuning the exploration and exploitation degree during the search process. In nature, the nymph grasshoppers have no wings, so they locally search for foods by moving in the surrounding area. However, adult grasshoppers fly freely in the air and explore a much larger scale region. In population-based optimisation techniques, however, exploring the search space comes first; due to the need for finding promising areas. Later on, the exploitation process is applied to locally intensify the search for finding better solutions that lead to the global optimum. For controlling the degree of exploratory and exploitative behaviours in GOA, the parameter c should be decreased proportionally to the number of executed iterations. This mechanism increases the degree of exploitation as the iteration count increases. It reduces the comfort zone proportionally to the number of iterations as well. Parameter c is calculated as in Eq. 7:

$$c = c_{Max} - l \frac{c_{Max} - c_{Min}}{L} \tag{7}$$

where c_{Max} is the maximum value of parameter c, c_{Min} is the minimum value of parameter c, l is the current iteration number, and L is the maximum number of iterations. In this work, we use

0.00001 and 1 for c_{Min} and c_{Max}, respectively, as recommended in the original GOA paper.

Algorithm 1 shows the pseudocode of the GOA algorithm. At

Algorithm 1 Pseudocode of the GOA algorithm.

Initialize c_{Max}, c_{Min}, and Max_Iterations
Initialize a population of solutions $X_i(i = 1, 2, \ldots, n)$
Evaluate each solution in the population
Set **T** as the best solution
while $t < Max$_Iterations **do**
    Update c using Eq. 7
    for each solution **do**
        Normalize the distances between grasshoppers in [1, 4]
        Update the location of the current solution using Eq. 6
        Bring the current grasshopper back if it goes outside the boundaries
    Update **T** if there is a better solution in population
    $t = t + 1$
return T

---

the beginning of the search process, GOA creates a set of random initial solutions (grasshoppers) and calculates the fitness for each of them. The grasshoppers update their positions based on Eq. (6). In each iteration, the position of the best grasshopper (target) that is obtained so far is updated. Furthermore, in each iteration, the parameter c is calculated using Eq. (7) and the distances between grasshoppers are normalised between 1 and 4. Updating the position of grasshoppers is performed iteratively until the stopping criterion is satisfied. Finally, the best grasshopper (target) is returned which represents the best approximation for the global optimum.

3. The proposed binary GOA (BGOA)

As mentioned in Saremi et al. (2017), GOA shows superior performance in solving optimisation problems compared to other optimisation methods in the literature. GOA shows very competitive performance concerning the quality of exploitation, exploration, local optima avoidance and convergence speed. The repulsion rate between grasshoppers in GOA plays a vital role in preventing the algorithm from being trapped in local optima, as well as in improving the exploration; since it requires individuals to avoid each other and extensively explore the search space. Besides that, the attraction forces between the individuals and the adaptive comfort zone are the two main factors that encourage the exploitation and convergence of the GOA algorithm. The adaptive behaviour of the comfort zone coefficient also results in a proper balance between exploration and exploitation (Saremi et al., 2017).

Searching for the best feature subset in FS is a challenging problem, especially in the wrapper-based methods. This is because the selected subset needs to be evaluated by the learning algorithm (e.g., classifier) at each individual optimisation step. Hence, a proper optimisation method is required to reduce the number of evaluations. The advantages of GOA mentioned above have motivated us to propose the use of this algorithm as a search method in a wrapper-based FS process. Based on the nature of FS problems, where the search space can be represented by binary values [0, 1], and since the binary operator are expected to be much simpler than continuous counterparts, we proposed a binary version of the GOA to solve the FS problem.

In the continuous version of GOA, each solution is updated based on its current position, the position of the best grasshopper so far (target), and the position of all other grasshoppers as in Eq. (6). The new solution produced by Eq. (6) is formulated by adding the step vector to the target (position vector). However, in a binary space, the add operator cannot be applied since

the position vector contains only0′ s and1′ s. Therefore, we proposed two approaches that aim to convert the continuous version of GOA to a binary one. The first approach uses the transfer functions (BGOA-S / BGOA-V), whereas the second uses the idea of mutation (BGOA / BGOA-M). The following three subsections explain these approaches in detail.

3.1. Binary grasshopper optimisation algorithm - Approach 1 (BGOA-S / BGOA-V)

In GOA, the first term of Eq. (6), redefined in Eq. (8) as ΔX, is analogous to the velocity vector (step) in PSO. According to Mirjalili and Lewis (2013), using transfer functions is one of the easiest ways to convert an algorithm from continuous to binary without modifying the structure. The transfer function defines the probability of updating the binary solution's elements from 0 to 1 and *vice versa*. In BGOA, the probability of changing positions elements is based on the step vector values.

$$\Delta X = c_1 \left(\sum_{j=1, \, j \neq i}^{N} c_2 \frac{ub_d - lb_d}{2} s\left(|x_j^d - x_i^d|\right) \frac{x_j - x_i}{d_{ij}} \right) \qquad (8)$$

Transfer functions are categorised into two families according to their shapes: S-shaped and V-shaped (Mirjalili & Lewis, 2013). Fig. 2 shows these two families of transfer functions.

Sigmoidal (S-shaped) function is a common transfer function, (see Eq. (9)) that was proposed by Kennedy and Eberhart (1997).

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

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

The position of the current grasshopper will be updated as in Eq. (10) based on the probability value $T_{(\Delta X)}$ obtained from Eq. (9).

$$X_{t+1}^d = \begin{cases} 1 & \text{If } r_1 < T\left(\Delta X_{t+1}^d\right) \\ 0 & \text{If } r_1 \geq T\left(\Delta X_{t+1}^d\right) \end{cases} \qquad (10)$$

where X_{t+1}^d represents the dth dimension of the grasshopper in the next iteration, and r_1 is a random number in the range [0, 1].

Hyperbolic tan (V-shaped) function is another transfer function (see Eq. (11)), that was proposed by Rashedi et al. (2010).

$$T(\Delta X_t) = |\tanh(\Delta X_t)| \qquad (11)$$

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

The position of the current grasshopper will be updated as in Eq. (12) based on the probability value $T(\Delta X)$ obtained from Eq. (11).

$$X_{t+1}^d = \begin{cases} \neg X_t^d & r_2 < T\left(\Delta X_{t+1}^d\right) \\ X_t^d & r_2 \geq T\left(\Delta X_{t+1}^d\right) \end{cases} \qquad (12)$$

where r_2 is a random number in the range [0, 1].

3.2. Binary grasshopper optimisation algorithm - Approach 2 (BGOA / BGOA-M)

In this approach, each grasshopper changes its position considering the locations of other grasshoppers (first term of Eq. (6)) and with respect to the location of the best solution found so far (Target). The attraction between the current grasshopper and the target can be represented by a stochastic mutation. The mutation rate is changing depending on the step vector itself. The positive elements (within the boundaries of the binary search space) will take the values of the corresponding elements (dimension) of the target; otherwise, to bring back the grasshoppers inside the search
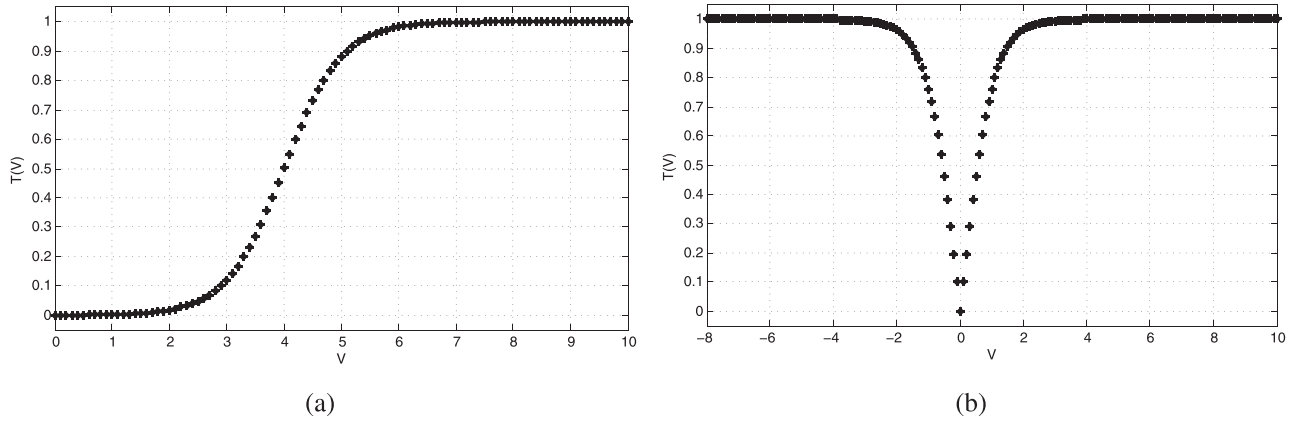
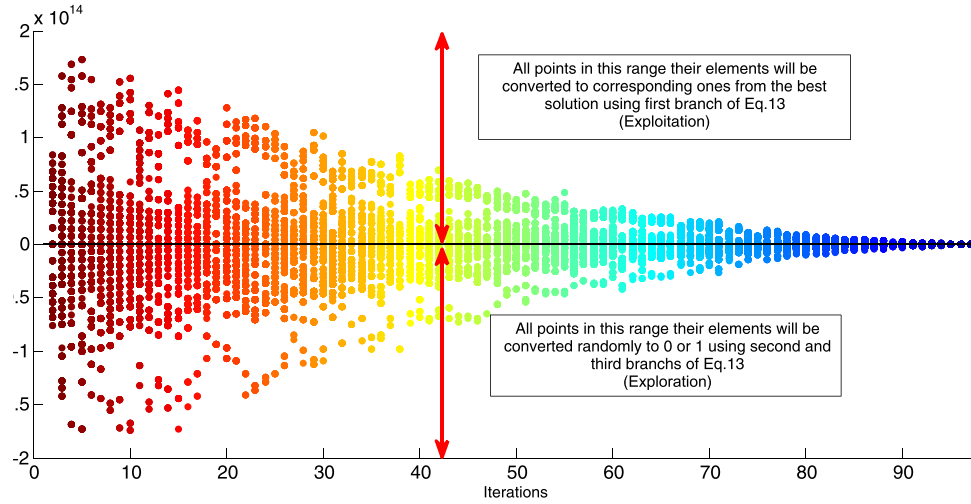**Fig. 2.** Transfer functions families (a) S-shaped and (b) V-shaped.

Fig. 3. Transfer mechanism in BGOA.

space, we randomly set this element based on a probability of 50% as presented in Eq. (13).

$$X_{t+1}^d = \begin{cases} Target_t^d & \Delta X_t^d \geq 0 \\ \begin{cases} 1 & r_3 \geq 0.5 \\ 0 & r_3 < 0.5 \end{cases} & \Delta X_t^d < 0 \end{cases} \quad (13)$$

where X_{t+1}^d represents the dth dimension of the grasshopper in the next iteration, ΔX_t^d is the value of the dth dimension in the step vector that is produced by Eq. (8), and r_3 is a random number in the range [0,1]. This mechanism is illustrated in Fig. 3. The first branch of Eq. (13) is responsible for the exploitation process, while the second and third branches are responsible for performing abrupt changes in the solution, which consequently means exploring different areas of the search space. The pseudocode of this approach is illustrated in Algorithm 2.

Then, the produced solution will be updated by performing mutation as a local search around new position with suitable mutation rate.

In the proposed BGOA approach, solutions are required to move towards the best solution so far. To avoid the premature convergence, a mutation operator with suitable mutation rate can be used to enhance the diversity of the algorithm. This approach is called BGOA-M. The mutation rate is a crucial factor that controls the mutation operator; as too high mutation rate increases the probability of searching more areas in search space, however, prevents population to converge to any optimal solutions. On the other hand, too

---

Algorithm 2 Pseudocode of the BGOA algorithm.

Initialize c_{Max}, c_{Min}, and $Max\_Iterations$
Initialize a population of solutions $X_i (i = 1, 2, \ldots, \dim)$
Evaluate each solution in the population
Set **Target** as the best solution
while $t < Max\_Iterations$ **do**
    Update c using Eq. 7
    for each solution **do**
        Normalize the distances between grasshoppers in [1, 4]
        Update the step vector ΔX_t of the current solution using Eq. 8
    for $i = 1 : \dim$ **do**
        if $\Delta X_t(i) > 0$ **then**
            $X_{t+1}(i) = TargetPosition(i)$;
        else
            if $rand >= 0.5$ **then**
                $X_{t+1}(i) = 1$;
            else
                $X_{t+1}(i) = 0$;
    $t = t + 1$
return Target

---

small mutation rate may result in premature convergence (falling in local optima instead of global optimum). The mutation rate r used in this paper is given in Eq. (14). The parameter r is linearly
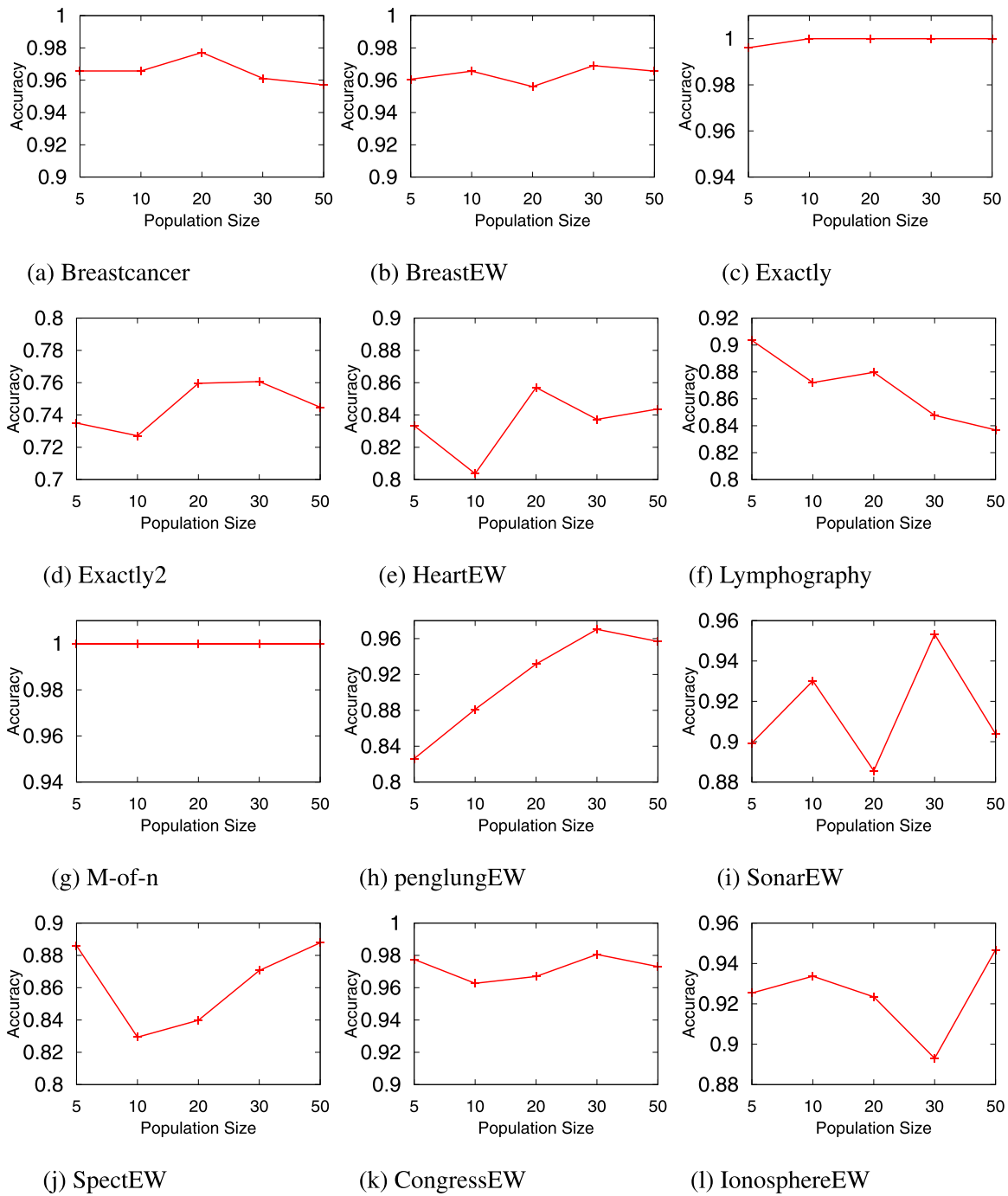
Fig. 4. Accuracy results of BGOA with different population sizes for Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, SonarEW, SpectEW, CongressEW and IonosphereEW datasets.

decremented from 0.9 to 0 depending on the iteration number i.

$$r = 0.9 + \frac{-0.9 * (i - 1)}{Max\_Iteration - 1} \qquad (14)$$

3.3. Fitness function

In the previous subsections, the importance of having an efficient searching strategy to the FS methods was shown. Another aspect of FS methods is how to assess the quality of the selected subset. Since the proposed approach is a wrapper-based method, then a learning algorithm (e.g., classifier) should be involved in the evaluation process. In this work, a well-known classifier; i.e.,

the k-Nearest Neighbor (k-NN) classifier (Altman, 1992), is utilised as an evaluator, and the classification accuracy of the selected features is involved in the proposed fitness function. When the selected features in a subset are relevant, the obtained classification accuracy will be better. Having a higher classification accuracy is one objective of FS methods, the other important objective is reducing the number of selected attributes, where the smaller the number of features in the solution, the better the solution is. In the proposed fitness function, these two contradictory objectives are taken into consideration. Eq. (15) shows the fitness function that considers both classification accuracy and the number of selected features when evaluating a feature subset of all approaches.

Fig. 5. Accuracy results of BGOA with different population sizes for KrvskpEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Zoo, Pendigits, Protein, Satellite, Spambase, Dermatology, Im, and Vehicle datasets.
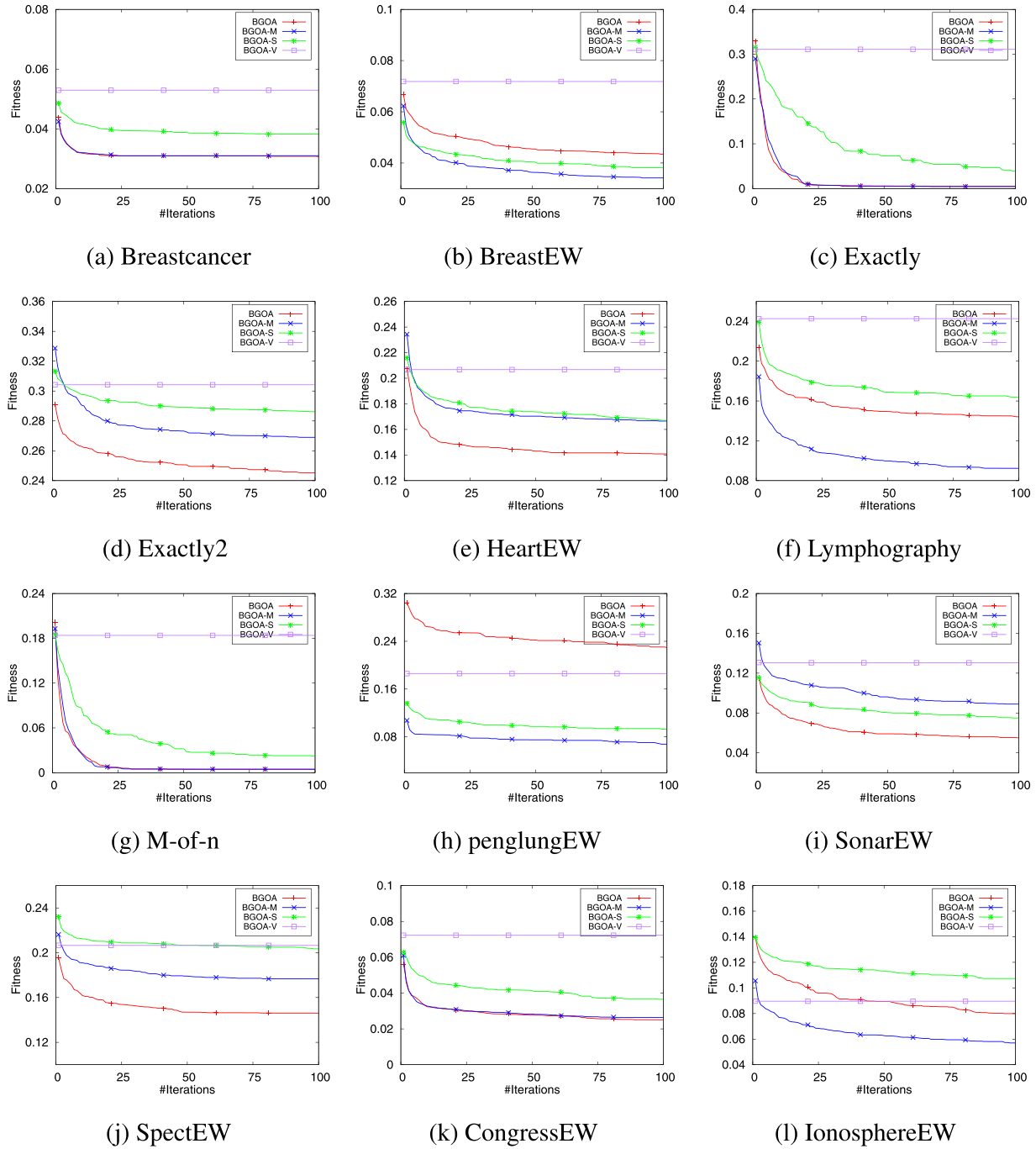
Fig. 6. Convergence curves of the proposed approaches for Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, SonarEW, SpectEW, CongressEW and IonosphereEW datasets.
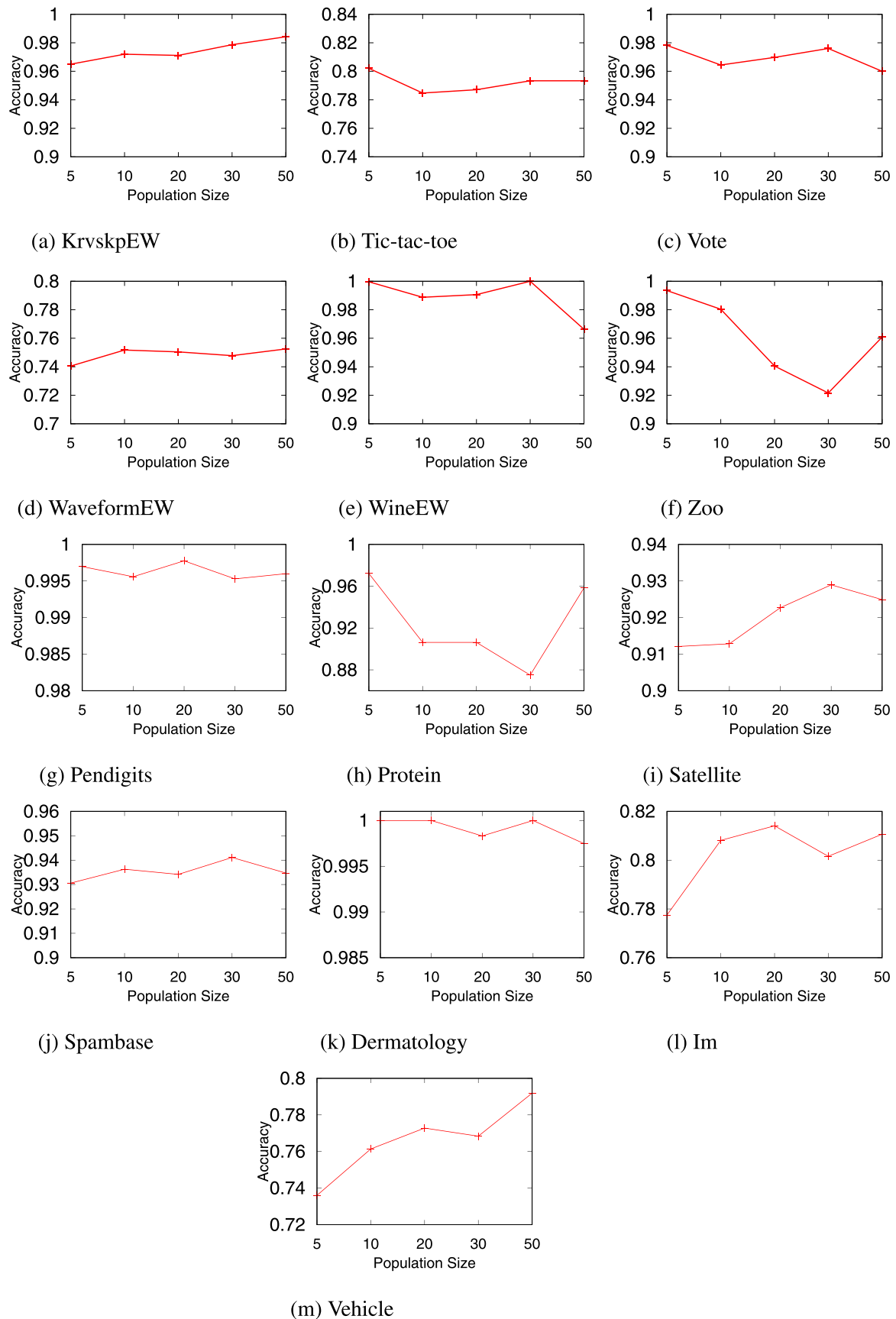
Since the number of selected features is to be minimised, we used the classification error rate (the complement of the classification accuracy).

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \qquad (15)$$

where $\gamma_R(D)$ is the classification error rate of the known classier, $|R|$ is the number of selected features, $|N|$ is the number of features, α and β are two parameters to reflect the role of classification rate and length of subset, and $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ are adopted from Emary, Zawbaa and Hassanien (2016b).

4. Experiments and results

Twenty-five different benchmark datasets from the UCI data repository (Dua & Karra Taniskidou, 2017) are utilised to test the proposed approaches. The datasets are described in terms of number of features, number of instances, number of classes, and dataset area in Table 1.

To obtain statistically meaningful results, 30 independent runs are performed for each approach. The statistical results concerning average fitness values, average classification accuracies, and the number of selected features for each approach along with their standard deviations are calculated. Convergence curves for all approaches are reported as well. All algorithms are implemented in
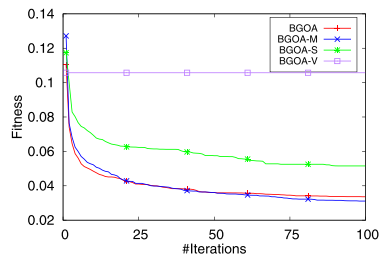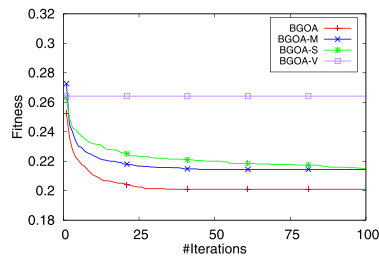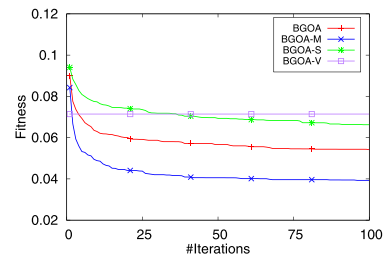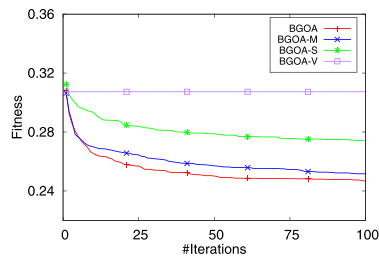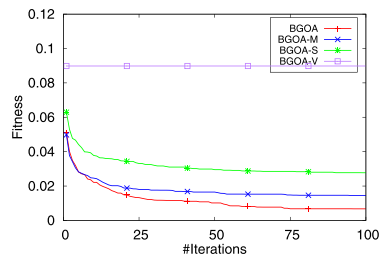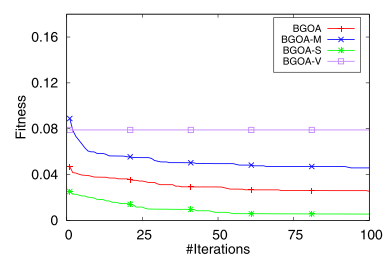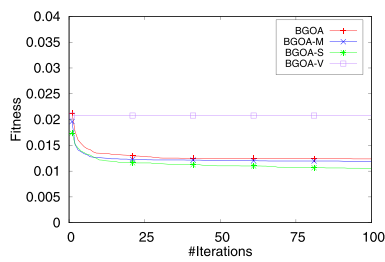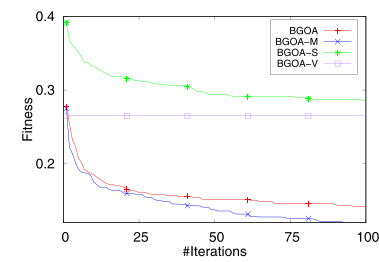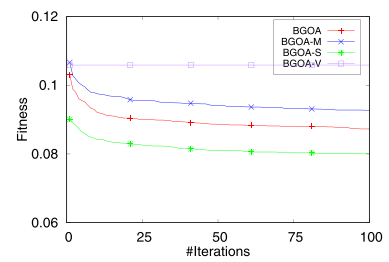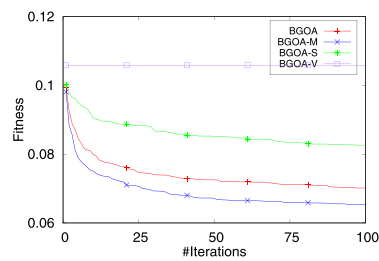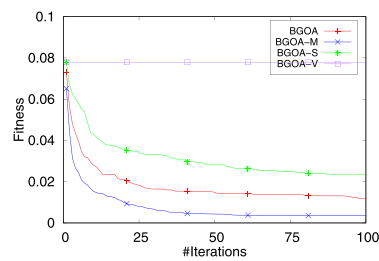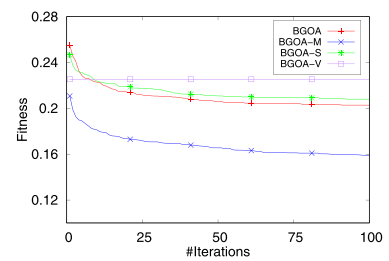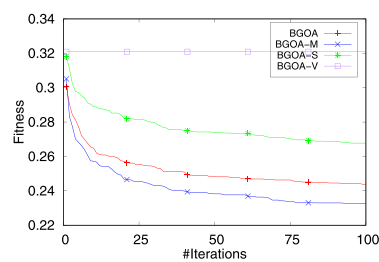
Fig. 7. Convergence curves for the proposed approaches for KrvskpEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Zoo, Pendigits, Protein, Satellite, Spambase, Dermatology, Im, and Vehicle datasets.
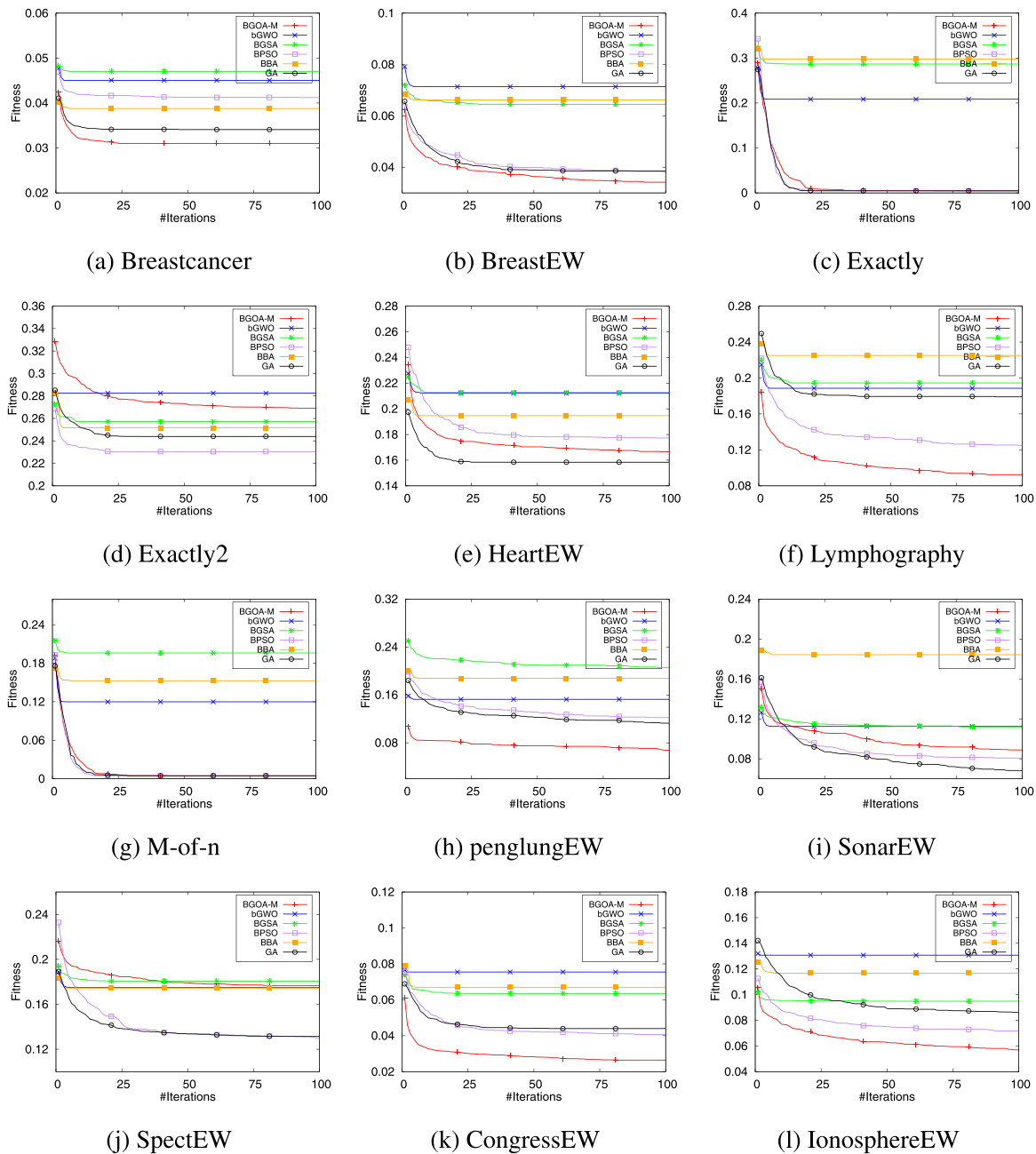
Fig. 8. Convergence curves for BGOA-M and other state-of-art methods for Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, SonarEW, SpectEW, CongressEW and IonosphereEW datasets.

Matlab and executed on an Intel Core i5 machine, 2.2 GHz CPU and 4 GB of RAM. The maximum number of iterations is 100. In this work, the k-NN classifier (where $k = 5$ (Emary, Zawbaa & Hassanien, 2016a) is used as an evaluator in the wrapper FS approach. In each of the 30 runs, each dataset is randomly divided into two sets; 80% of the instances are used for training, and the remaining are used for testing. This partitioning was used in various previous works in the literature (Alsaafin & Elnagar, 2017; Li, Shrivastava, Moore & König, 2011; Wieland & Pittore, 2014).

As a preliminary study, we experimented the effect of the population size on the performance of the basic approach, which is BGOA, in terms of classification accuracy. Therefore, the BGOA was evaluated at different population sizes (*i.e.,* 5, 10, 20, 30 and 50). Figs. 4 and 5 show the impact of changing the population size on

the attained classification accuracy for all datasets. It can be noticed that the range of variation in the classification accuracy in most of the datasets is minimal. In addition, increasing the population size does not always improve the results. Therefore, we set the population size to 10 for all next experiments as a trade-off between the classification accuracy and the overhead of the running time of the algorithm.

To verify the effectiveness of the proposed approaches, the experiments are conducted based on four stages: first, the performance of the proposed approaches are evaluated based on the datasets as mentioned earlier. Then, the best performing approach is compared with the state-of-art metaheuristic methods. After that, a comparison with metaheuristics in the literature is reported. Finally, the best GOA-based method is compared against different
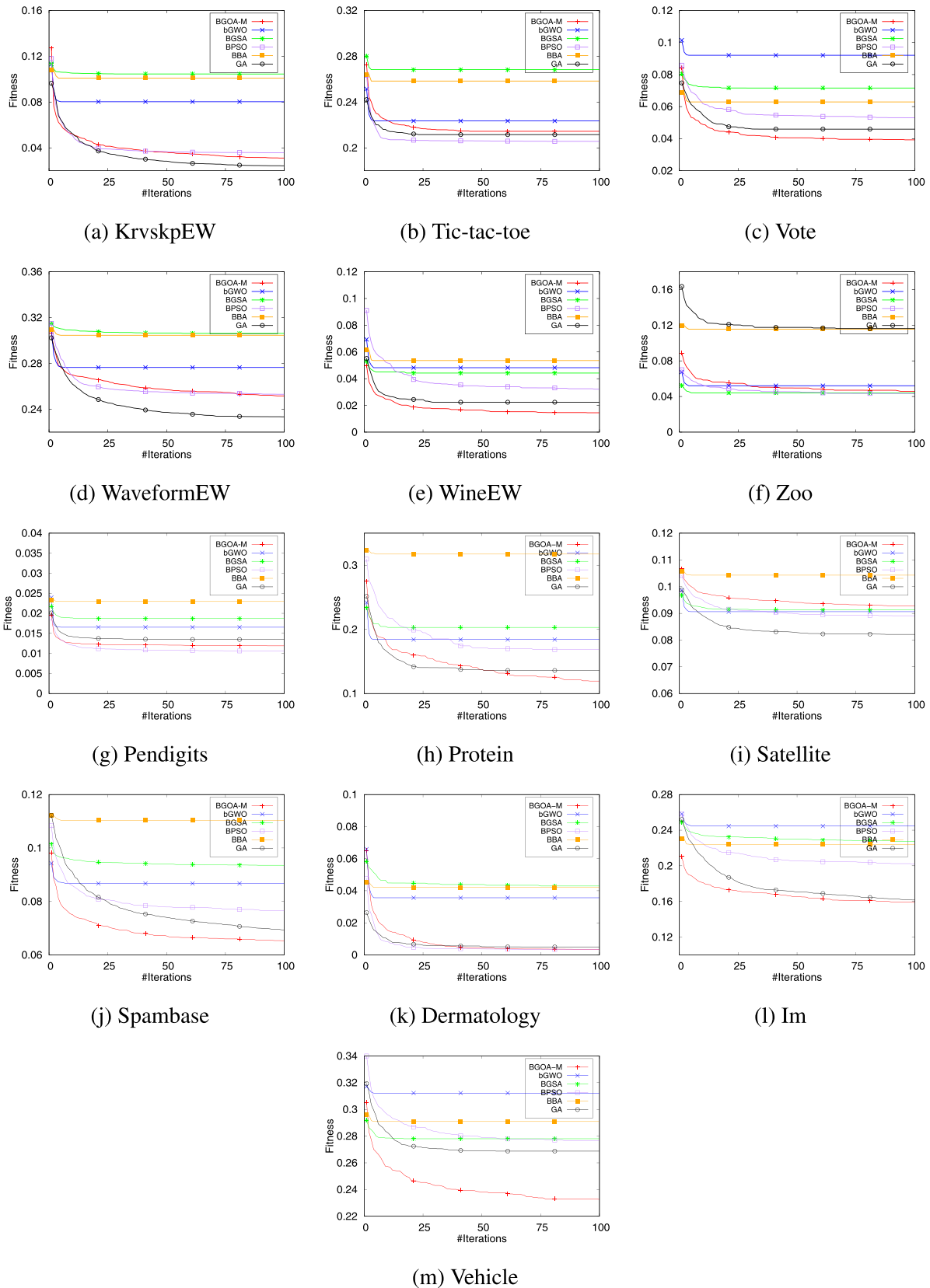
Fig. 9. Convergence curves for BGOA-M and other state-of-art methods for KrvskpEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Zoo, Pendigits, Protein, Satellite, Spambase, Dermatology, Im, Vehicle datasets.

Table 1
List of datasets used in the experiments.

No.	Dataset	No. offeatures	No. of instances	No. of classes	Data area
1.	Breastcancer	9	699	2	Biology
2.	BreastEW	30	569	2	Biology
3.	Exactly	13	1000	2	Biology
4.	Exactly2	13	1000	2	Biology
5.	HeartEW	13	270	2	Biology
6.	Lymphography	18	148	2	Biology
7.	M-of-n	13	1000	2	Biology
8.	PenglungEW	325	73	2	Biology
9.	SonarEW	60	208	2	Biology
10.	SpectEW	22	267	2	Biology
11.	CongressEW	16	435	2	Politics
12.	IonosphereEW	34	351	2	Electromagnetic
13.	KrvskpEW	36	3196	2	Game
14.	Tic-tac-toe	9	958	2	Game
15.	Vote	16	300	2	Politics
16.	WaveformEW	40	5000	3	Physics
17.	WineEW	13	178	3	Chemistry
18.	Zoo	16	101	6	Artificial
19.	Pendigits	16	7494	10	Handwriting
20.	Protein	20	116	6	Biology
21.	Satellite	35	6435	6	Physical
22.	Spambase	57	4601	2	Computer
23.	Dermatology	34	366	6	Biology
24.	Im	270	1000	10	Business
25.	Vehicle	18	846	4	Artificial

Table 2
Comparison between the proposed approaches based on average classification accuracy (AvgAcc).

| Benchmark	BGOA		BGOA-M		BGOA-S		BGOA-V	
	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev
Breastcancer	0.9737	0.0012	**0.9743**	**0.0000**	0.9678	0.0013	0.9519	0.0057
BreastEW	0.9606	0.0042	**0.9697**	0.0040	0.9670	0.0036	0.9327	0.0068
Exactly	0.9999	0.0007	**1.0000**	**0.0000**	0.9667	0.0300	0.6915	0.0747
Exactly2	**0.7565**	0.0103	0.7352	0.0023	0.7163	0.0040	0.6973	0.0248
HeartEW	**0.8635**	0.0054	0.8358	**0.0028**	0.8363	0.0071	0.7968	0.0192
Lymphography	0.8586	0.0105	**0.9118**	0.0114	0.8392	0.0089	0.7599	0.0317
M-of-n	**1.0000**	**0.0000**	**1.0000**	**0.0000**	0.9830	0.0196	0.8203	0.0658
penglungEW	0.7712	0.0197	**0.9342**	0.0154	0.9108	0.0126	0.8171	0.0154
SonarEW	**0.9497**	0.0074	0.9147	0.0109	0.9295	0.0081	0.8734	0.0168
SpectEW	**0.8565**	0.0072	0.8261	0.0076	0.7983	0.0054	0.7963	0.0139
CongressEW	**0.9772**	0.0031	0.9764	**0.0016**	0.9665	0.0042	0.9315	0.0140
IonosphereEW	0.9220	0.0076	**0.9458**	0.0073	0.8955	0.0059	0.9144	0.0109
KrvskpEW	0.9708	0.0038	**0.9736**	0.0030	0.9533	0.0068	0.8989	0.0473
Tic-tac-toe	**0.8038**	**0.0000**	0.7912	**0.0000**	0.7894	0.0043	0.7392	0.0335
Vote	0.9484	0.0039	**0.9633**	0.0034	0.9364	0.0045	0.9327	0.0123
WaveformEW	**0.7562**	0.0049	0.7511	0.0064	0.7289	0.0076	0.6951	0.0148
WineEW	**0.9985**	0.0039	0.9888	**0.0000**	0.9775	0.0042	0.9146	0.0149
Zoo	0.9778	0.0161	0.9575	0.0074	**1.0000**	**0.0000**	0.9261	0.0183
Pendigit	0.9954	0.0007	0.9956	**0.0003**	**0.9968**	0.0006	0.9858	0.0053
Protein	0.8611	**0.0200**	**0.8841**	0.0305	0.7167	0.0202	0.7375	0.0581
Satelite	0.9180	0.0030	0.9120	**0.0020**	**0.9250**	0.0020	0.8990	0.0040
Spambase	0.9349	**0.0041**	**0.9398**	0.0041	0.9219	0.0036	0.8982	0.0113
Dermatology	0.9923	0.0068	**1.0000**	**0.0000**	0.9806	0.0068	0.9266	0.0212
Im	0.7998	**0.0065**	**0.8445**	0.0083	0.7948	0.0076	0.7775	0.0153
Vehicle	0.7582	0.0144	**0.7704**	0.0115	0.7345	**0.0072**	0.6812	0.0202
Average Accuracy	0.9042	0.0066	0.9118	0.0056	0.8893	0.0074	0.8398	0.0230
Sum of the ranks	45		40		67		97	
Average rank	1.8		1.6		2.68		3.88	
Final rank	2		1		3		4	

filter-based FS methods. The average rankings of the proposed approaches in terms of Accuracy, number of selected features, and fitness metrics are presented at the bottom of each table.

4.1. Comparison between the GOA based approaches

This subsection presents the results obtained by the four proposed approaches. The proposed algorithms are quantitatively compared using the following metrics:

- The Classification accuracy using the selected features on the test dataset. The average classification accuracy and the standard deviation are calculated for the 30 runs.
- The average and the standard deviation of the selected size of the features subset.
- Convergence curves of the best-so-far solutions.

The results in terms of the average classification accuracy are shown in Table 2. As it can be seen, BGOA-M has achieved the best results where it succeeded to hit the highest accuracy in four-

Table 3
Comparison between the proposed approaches based on average number of selected features (AvgNF).

Benchmark	BGOA		BGOA-M		BGOA-S		BGOA-V	
	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev
Breastcancer	**4.2000**	0.4068	5.0000	**0.0000**	5.7333	0.4498	4.8000	0.8469
BreastEW	13.2667	2.6121	**12.5000**	**1.9783**	16.2333	2.2997	15.8000	2.8089
Exactly	6.0333	0.1826	**6.0000**	**0.0000**	7.4000	0.6747	7.0333	2.4980
Exactly2	**5.4000**	2.2221	8.9333	**0.3651**	6.9667	2.0424	5.8667	2.1453
HeartEW	7.4333	1.1043	**5.2000**	**0.5509**	6.5667	1.3817	7.2333	1.7157
Lymphography	**7.4000**	1.4994	9.0333	**1.3257**	8.6333	2.6325	9.0667	1.9989
M-of-n	**6.0000**	**0.0000**	**6.0000**	**0.0000**	7.3667	0.8899	7.8000	1.2972
penglungEW	95.7000	18.1053	86.1000	24.8379	152.1333	**8.4107**	157.7667	9.2015
SonarEW	30.5333	4.2729	**26.8000**	4.2378	30.8000	**3.4581**	30.3000	3.9580
SpectEW	8.7333	2.5042	9.9667	1.3515	**8.2667**	**1.1121**	10.7333	2.3625
CongressEW	**4.0667**	2.6901	5.0000	1.2594	5.4667	**1.1666**	7.1333	1.8144
IonosphereEW	**9.1333**	2.8129	11.4667	2.7510	13.1000	2.6826	16.4667	**2.5015**
KrvskpEW	**16.9333**	2.8999	17.7333	3.3315	19.0000	2.4212	20.2000	**1.8644**
Tic-tac-toe	6.0000	**0.0000**	7.0000	**0.0000**	5.8667	0.3457	**5.4333**	1.1651
Vote	4.9667	0.3198	**4.7667**	1.5906	5.3667	1.2172	7.6333	2.1891
WaveformEW	21.5667	3.1697	**20.9000**	**2.9048**	22.3667	3.6054	21.6333	3.1457
WineEW	6.8333	**0.6477**	**4.4000**	1.8118	7.1333	1.5025	6.8667	1.3322
Zoo	**5.4333**	1.2229	6.0000	**0.5872**	8.8667	0.8193	9.3000	1.8033
Pendigit	12.4667	1.2521	12.1333	**0.3457**	11.8000	1.0954	**10.7333**	1.0807
Protein	**8.3000**	2.0197	9.4667	1.9429	10.4000	**1.5222**	10.5667	1.7357
Satelite	23.4670	3.0370	20.2000	2.2500	21.1670	**1.8210**	**19.6670**	2.4960
Spambase	32.1667	3.6586	32.7000	4.5040	30.2000	3.6897	**28.7333**	**3.3521**
Dermatology	14.2333	4.7611	**11.9333**	**2.3034**	15.2667	2.3916	17.8000	2.7593
Im	**125.6000**	20.1316	137.6000	18.5595	133.0333	**6.5310**	138.0667	7.2299
Vehicle	**8.0333**	**0.9994**	9.6000	1.3287	8.8667	1.6761	9.5333	2.1453
Average Features	19.3560	3.3013	0, 0, 119.4573	3.2047	22.7200	2.2336	23.4467	2.6179
Sum of the ranks	47		53		65		76	
Average rank	1.88		2.12		2.6		3.04	
Final rank	1		2		3		4	

Table 4
Comparison between the proposed approaches based on average of fitness (AvgF).

Benchmark	BGOA		BGOA-M		BGOA-S		BGOA-V	
	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev
Breastcancer	**0.0307**	0.0016	0.0310	**0.0000**	0.0382	0.0008	0.0529	0.0058
BreastEW	0.0434	0.0039	**0.0342**	0.0037	0.0381	**0.0034**	0.0718	0.0065
Exactly	0.0048	0.0009	**0.0046**	**0.0000**	0.0386	0.0302	0.3109	0.0736
Exactly2	**0.2452**	0.0086	0.2690	0.0020	0.2862	0.0040	0.3042	0.0257
HeartEW	**0.1409**	0.0048	0.1666	**0.0031**	0.1671	0.0066	0.2067	0.0188
Lymphography	0.1441	0.0103	**0.0924**	0.0111	0.1640	**0.0093**	0.2427	0.0313
M-of-n	**0.0046**	**0.0000**	0.0046	**0.0000**	0.0225	0.0199	0.1839	0.0651
penglungEW	0.2295	0.0195	**0.0678**	0.0149	0.0930	**0.0123**	0.1859	0.0151
SonarEW	**0.0549**	**0.0073**	0.0889	0.0105	0.0749	0.0079	0.1304	0.0166
SpectEW	**0.1461**	0.0067	0.1767	0.0074	0.2035	**0.0052**	0.2066	0.0143
CongressEW	**0.0251**	0.0034	0.0264	**0.0013**	0.0366	0.0045	0.0723	0.0143
IonosphereEW	0.0799	0.0079	**0.0570**	0.0074	0.1074	**0.0059**	0.0896	0.0107
KrvskpEW	0.0336	0.0037	**0.0311**	**0.0030**	0.0515	0.0066	0.1057	0.0468
Tic-tac-toe	**0.2010**	0.0038	0.2145	**0.0000**	0.2151	0.0040	0.2642	0.0327
Vote	0.0541	0.0038	**0.0393**	0.0024	0.0663	0.0044	0.0714	0.0130
WaveformEW	**0.2467**	**0.0052**	0.2516	0.0067	0.2740	0.0075	0.3073	0.0145
WineEW	**0.0067**	0.0035	0.0145	**0.0014**	0.0277	0.0038	0.0898	0.0146
Zoo	0.0254	0.0153	0.0458	0.0071	**0.0055**	**0.0005**	0.0789	0.0178
Pendigit	0.0124	**0.0003**	0.0119	0.0003	0.0105	0.0007	0.0208	0.0048
Protein	0.1417	**0.0193**	**0.1195**	0.0302	0.2857	0.0200	0.2652	0.0573
Satelite	0.0870	**0.0020**	0.0930	0.0020	**0.0800**	0.0020	0.1060	0.0040
Spambase	0.0701	0.0039	**0.0653**	0.0038	0.0826	**0.0034**	0.1059	0.0110
Dermatology	0.0118	0.0056	**0.0035**	**0.0007**	0.0237	0.0066	0.0779	0.0212
Im	0.2028	**0.0061**	**0.1590**	0.0082	0.2080	0.0075	0.2254	0.0152
Vehicle	0.2438	0.0145	**0.2327**	0.0111	0.2678	**0.0071**	0.3209	0.0202
Average Fitness	0.0995	0.0063	0.0920	0.0055	0.1147	0.0074	0.1639	0.0228
Sum of the ranks	44		40		68		97	
Average rank	1.76		1.6		2.72		3.88	
Final rank	2		1		3		4	

teen datasets. BGOA comes in the second place where it could outperform other approaches in nine datasets. BGOA-S comes next by achieving the best results in three datasets which are Zoo, Pindigits, and Satellite datasets. BGOA-V comes in the last place, where it could not outperform the other methods in any dataset. Moreover, the latter method showed low classification accuracy for some datasets compared to the other methods like in Exactly, Lymphography, and SonarEW. Inspecting the standard deviation for each method, we can see that BGOA-M has the smallest values in fourteen datasets, while BGOA comes next with six datasets. The small standard deviation values indicate the stability of the algorithms.

Table 5
Best obtained results of BGOA, BGOA-M, BGOA-S, and BGOA-V concerning fitness values (F), classification accuracy (Acc) and number of selected features (NF).

Benchmark	BGOA			BGOA-M			BGOA-S			BGOA-V		
	F	Acc	NF	F	Acc	NF	F	Acc	NF	F	Acc	NF
Breastcancer	**0.0299**	**0.9743**	4	0.0310	**0.9743**	5	0.0378	0.9686	5	0.0412	0.9629	**3**
BreastEW	0.0349	0.9684	**8**	0.0286	0.9754	10	**0.0290**	**0.9754**	13	0.0568	0.9474	11
Exactly	**0.0046**	**1.0000**	6	0.0046	1.0000	6	0.0046	1.0000	6	0.0557	0.9500	**1**
Exactly2	**0.2326**	**0.7720**	4	0.2683	0.7360	7	0.2774	0.7260	5	0.2451	0.7540	**2**
HeartEW	**0.1382**	**0.8667**	4	0.1652	0.8370	5	0.1586	0.8444	**3**	0.1748	0.8296	**3**
Lymphography	0.1371	0.8649	4	**0.0847**	**0.9189**	7	0.1371	0.8649	**3**	0.1371	0.8649	6
M-of-n	**0.0046**	**1.0000**	6	0.0046	1.0000	6	0.0046	1.0000	6	0.0046	1.0000	**5**
penglungEW	0.1635	0.8378	64	**0.0287**	**0.9730**	**36**	0.0846	0.9189	135	0.1390	0.8649	139
SonarEW	**0.0426**	**0.9615**	23	0.0716	0.9327	**16**	0.0623	0.9423	21	0.0903	0.9135	23
SpectEW	**0.1306**	**0.8731**	**4**	0.1602	0.8433	7	0.1888	0.8134	7	0.1805	0.8209	6
CongressEW	**0.0200**	**0.9817**	**2**	0.0252	0.9771	3	0.0252	0.9771	4	0.0382	0.9633	3
IonosphereEW	0.0636	0.9375	**4**	**0.0364**	**0.9659**	7	0.0932	0.9091	8	0.0669	0.9375	11
KrvskpEW	0.0272	0.9775	**12**	**0.0259**	**0.9800**	11	0.0419	0.9625	14	0.0427	0.9625	16
Tic-tac-toe	**0.2010**	**0.8038**	6	0.2145	0.7912	7	0.2134	0.7912	5	0.2175	0.7871	**3**
Vote	0.0493	0.9533	4	**0.0368**	**0.9667**	**3**	0.0500	0.9533	4	0.0434	0.9600	**3**
WaveformEW	**0.2340**	**0.7684**	16	0.2415	0.7604	**14**	0.2567	0.7460	16	0.2696	0.7320	16
WineEW	**0.0054**	**1.0000**	5	0.0134	0.9888	**3**	0.0180	0.9888	5	0.0706	0.9326	4
Zoo	**0.0038**	**1.0000**	4	0.0426	0.9608	5	0.0044	**1.0000**	7	0.0444	0.9608	5
Pendigit	0.0121	0.9960	11	0.0115	0.9960	12	**0.0089**	**0.9980**	10	0.0141	0.9940	**8**
Protein	0.1263	0.8750	**5**	**0.0511**	**0.9524**	6	0.2520	0.7500	8	0.1288	0.8750	8
Satelite	0.0829	0.9246	18	0.0881	0.9161	16	**0.0765**	**0.9285**	18	0.0992	0.9060	**15**
Spambase	0.0597	0.9446	25	**0.0566**	**0.9501**	26	0.0734	0.9305	23	0.0830	0.9218	**22**
Dermatology	0.0041	1.0000	8	**0.0024**	**1.0000**	8	0.0166	0.9865	11	0.0440	0.9595	12
Im	0.1922	0.8100	**84**	**0.1448**	**0.8600**	100	0.1833	0.8200	119	0.1982	0.8050	123
Vehicle	**0.2077**	**0.7941**	**6**	0.2210	0.7824	7	0.2554	0.7471	**6**	0.2787	0.7235	**6**
Average	0.0883	0.9154	13.4800	0.0824	0.9215	13.3200	0.1021	0.9017	18.4800	0.1106	0.8931	18.1600
Sum of the ranks	44	41	47	43	42	60	63	60	66	85	84	53
Average rank	1.76	1.64	1.88	1.72	1.68	2.4	2.52	2.4	2.64	3.4	3.36	2.12
Final rank	2	1	1	1	2	3	3	3	4	4	4	2

Overall, BGOA-M shows the best accuracy results with higher stability.

In Table 3, the proposed GOA methods are compared to each other in terms of the average number of selected features. BGOA-M has the smallest average number of features in ten datasets, while BGOA has the smallest values in eleven datasets. BGOA-S and BGOA-V had the least number of features in one and three datasets, respectively. As for the standard deviation values, BGOA-M is shown to be a robust approach that it obtained the smallest values in half of the datasets. The superiority of BGOA-M is also confirmed by the average fitness results as shown in Table 4. BGOA-M has the best fitness values in fourteen datasets, while BGOA is best in nine datasets.

The average convergence curves of the four methods for all datasets are shown in Figs. 6 and 7. It can be seen that BGOA-M and BGOA show the fastest convergence in most of the datasets, while BOGA-V is the slowest to converge. It is worth mentioning that 100 iterations were enough for BGOA-M and BGOA to converge in all cases. Observing the best results of classification accuracy, fitness values and number of selected features in Table 5, we can see that BGOA and BGOA-M managed to hit the best rate in most of the datasets.

4.2. Comparison with other metaheuristic-based approaches

This subsection presents a comparison between BGOA-M, as the best approach among the proposed approaches in this work, and other state-of-the-art metaheuristics-based FS algorithms. The algorithms that are used for this comparison are BGWO, BGSA, BPSO, BBA, and GA, which are considered the most popular and well-regarded metaheuristic-based FS algorithms in the literature. In addition, and for baseline comparisons, the standard k-NN is used on the complete set of features.

Table 6
The parameter settings.

Algorithm	Parameter	Value
PSO	Individual-best acceleration factor	0.1
	c_1	2
	c_2	2
GSA	G_0	1
	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]
GA	Crossover	0.8
	Mutation	0.02
GOA	interval	[0 2.079]
	l	1.5
	f	0.5

All algorithms are quantitatively compared using the same metrics described in the preceding subsection. The parameters of optimizers are sensibly selected using many trial and error processes to comprehend the finest feasible settings. Table 6 shows the used parameters in the all experiments.

Table 7 presents the results of average classification accuracy of all approaches and k-NN baseline method that is applied on the complete set of features. We notice that the best results are dominated by BGOA-M, GA, and PSO. BGOA-M has the best performance, where it obtained the best classification results in 60% of the datasets (15 out of 25). GA and PSO come next by obtaining the best results in seven datasets. k-NN comes as the last method by obtaining the worst results. Examining the results in Table 7, we see that BGOA-M was able to outperform the other methods with a remarkable difference in some datasets such as in Lymphography and PenglungEW. The results also confirm the stability of the pro-

Table 7
Comparison between BGOA-M and state-of-art methods based on average classification accuracy (AvgAcc).

Benchmark	BGOA_M		bGWO		BGSA		BPSO		BBA		GA		k-NN
	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev	AvgAcc	StdDev	Acc
Breastcancer	**0.9743**	**0.0000**	0.9617	0.0031	0.9588	0.0050	0.9629	**0.0000**	0.9326	0.0528	0.9720	0.0031	0.9371
BreastEW	**0.9697**	0.0040	0.9349	0.0071	0.9399	0.0065	0.9656	**0.0034**	0.9196	0.0208	0.9655	0.0060	0.9368
Exactly	**1.0000**	**0.0000**	0.7975	0.0673	0.7166	0.0862	**1.0000**	**0.0000**	0.5989	0.0455	**1.0000**	**0.0000**	0.7120
Exactly2	0.7352	0.0023	0.7188	0.0224	0.7437	0.0266	**0.7680**	**0.0000**	0.6846	0.0421	0.7570	0.0151	0.6780
HeartEW	0.8358	**0.0028**	0.7926	0.0149	0.7911	0.0217	0.8247	0.0100	0.7563	0.0555	**0.8444**	0.0135	0.6593
Lymphography	**0.9118**	0.0114	0.8165	0.0267	0.8086	0.0225	0.8781	0.0128	0.7243	0.0723	0.8230	0.0152	0.7297
M-of-n	**1.0000**	**0.0000**	0.8868	0.0495	0.8080	0.0536	**1.0000**	**0.0000**	0.7585	0.1017	**1.0000**	**0.0000**	0.8100
penglungEW	**0.9342**	0.0154	0.8514	0.0210	0.7971	0.0193	0.8811	0.0168	0.7883	0.0266	0.8892	0.0205	0.8108
SonarEW	0.9147	0.0109	0.8929	0.0144	0.8923	0.0114	0.9231	**0.0087**	0.7593	0.0444	**0.9356**	0.0193	0.8365
SpectEW	0.8261	**0.0076**	0.8306	0.0126	0.8214	0.0114	**0.8739**	0.0111	0.7876	0.0378	0.8719	0.0147	0.7388
CongressEW	**0.9764**	0.0016	0.9283	0.0089	0.9402	0.0093	0.9612	0.0026	0.8656	0.0760	0.9592	0.0082	0.9266
IonosphereEW	**0.9458**	0.0073	0.8729	0.0138	0.9085	0.0146	0.9311	0.0074	0.8629	0.0280	0.9165	0.0171	0.8239
KrvskpEW	0.9736	0.0030	0.9258	0.0120	0.9001	0.0477	0.9693	**0.0022**	0.8152	0.0740	**0.9798**	0.0040	0.8986
Tic-tac-toe	0.7912	**0.0000**	0.7822	0.0163	0.7353	0.0211	**0.7990**	0.0009	0.6697	0.0593	0.7936	0.0082	0.7683
Vote	**0.9633**	0.0034	0.9140	0.0077	0.9320	0.0090	0.9509	0.0051	0.8669	0.0792	0.9580	0.0118	0.9067
WaveformEW	0.7511	0.0064	0.7291	0.0079	0.6958	0.0138	0.7494	**0.0040**	0.6610	0.0350	**0.7693**	0.0063	0.7004
WineEW	**0.9888**	**0.0000**	0.9584	0.0111	0.9607	0.0101	0.9727	0.0057	0.8861	0.0800	0.9820	0.0070	0.9326
Zoo	0.9575	0.0074	0.9536	0.0227	0.9608	0.0186	0.9601	**0.0036**	0.7915	0.1101	0.8856	0.0242	**0.9800**
Pendigit	0.9956	0.0003	0.9921	0.0010	0.9879	0.0046	**0.9973**	0.0006	0.9174	0.1494	0.9947	0.0014	0.9923
Protein	**0.8841**	0.0305	0.8208	0.0412	0.8000	0.0443	0.8347	**0.0204**	0.5764	0.0991	0.8667	0.0254	0.5862
Satelite	0.9120	**0.0020**	0.9160	0.0030	0.9130	0.0040	0.9160	**0.0020**	0.8890	0.0130	**0.9230**	0.0040	0.8987
Spambase	**0.9398**	0.0041	0.9192	0.0096	0.9106	0.0092	0.9281	**0.0026**	0.8672	0.0267	0.9347	0.0088	0.7984
Dermatology	**1.0000**	**0.0000**	0.9712	0.0122	0.9617	0.0147	**1.0000**	**0.0000**	0.9072	0.0604	0.9978	0.0051	0.8798
Im	**0.8445**	0.0083	0.7598	0.0128	0.7753	0.0107	0.8008	0.0093	0.7460	0.0203	0.8415	0.0162	0.7480
Vehicle	**0.7704**	0.0115	0.6906	0.0213	0.7245	0.0187	0.7251	**0.0109**	0.6659	0.0598	0.7333	0.0150	0.6619
Average Accuracy	0.9118	0.0056	0.8647	0.0176	0.8554	0.0206	0.9029	0.0056	0.7879	0.0588	0.9038	0.0108	0.8141
Sum of the ranks	46		105		117		55		167		53		146
Average rank	1.84		4.2		4.68		2.2		6.68		2.12		5.84
Final rank	1		4		5		3		7		2		6

Table 8
Comparison between BGOA-M and state-of-art methods based on average number of selected features (AvgNF).

Benchmark	BGOA_M		bGWO		BGSA		BPSO		BBA		GA	
	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev	AvgNF	StdDev
Breastcancer	5.0000	**0.0000**	6.4333	1.2229	5.6000	1.0700	**4.0000**	**0.0000**	4.3000	1.8597	5.7333	0.5833
BreastEW	12.5000	**1.9783**	20.6000	3.6351	15.0333	2.3995	12.6667	2.3538	**12.3333**	2.1549	13.2000	2.1877
Exactly	6.0000	**0.0000**	10.3333	1.4223	8.2333	1.3047	6.0000	**0.0000**	**5.7667**	1.6121	6.0000	**0.0000**
Exactly2	8.9333	0.3651	5.2667	2.3034	4.4000	2.5271	**1.0000**	**0.0000**	4.8667	1.8520	4.4000	3.8111
HeartEW	5.2000	**0.5509**	8.8333	1.4404	7.8000	1.8080	**4.8333**	1.5332	5.2000	1.5177	5.7000	0.8769
Lymphography	9.0333	1.3257	12.4333	2.8000	8.8667	1.6554	7.7000	2.2307	7.8333	2.2296	**6.7667**	**1.3047**
M-of-n	6.0000	0.0000	10.2000	1.3995	7.7000	1.1788	6.0000	**0.0000**	**5.9333**	2.1324	6.0000	**0.0000**
penglungEW	**86.1000**	24.8379	174.3333	30.4918	158.4667	8.7404	143.3333	**6.9448**	123.5667	19.3225	99.6667	9.2301
SonarEW	26.8000	4.2378	38.7667	6.9515	30.0667	3.3726	27.9667	**2.8099**	**23.5667**	4.6512	27.0667	2.9117
SpectEW	9.9667	**1.3515**	15.7333	3.8768	**8.4333**	2.2234	11.5667	1.5687	9.0333	2.2816	10.0000	2.3635
CongressEW	5.0000	1.2594	7.1667	1.7036	6.9333	1.6802	**3.6000**	**0.7240**	6.0333	2.4842	5.7333	1.9286
IonosphereEW	11.4667	2.7510	16.8667	3.2135	14.7333	2.6901	11.8667	**2.4316**	11.6333	3.3680	**11.0667**	2.7908
KrvskpEW	17.7333	3.3315	24.5333	4.7251	19.9000	3.2626	18.8333	**2.1985**	15.9667	3.0904	**15.7000**	3.7060
Tic-tac-toe	7.0000	**0.0000**	7.1000	1.6049	5.6000	1.0034	6.0000	**0.0000**	**4.1333**	1.1366	6.5000	0.5724
Vote	**4.7667**	1.5906	11.0333	3.0680	6.8333	1.5992	7.0000	3.2056	5.8667	2.2242	7.0000	**1.1142**
WaveformEW	20.9000	2.9048	33.1667	4.2270	20.0333	2.8465	20.1000	2.9752	**17.6333**	3.6054	20.2000	**2.5107**
WineEW	**4.4000**	1.8118	9.2000	1.9191	6.9000	1.7090	6.6000	**0.8137**	5.7333	1.7006	6.0000	1.3646
Zoo	6.0000	0.5872	9.8000	1.9010	8.3667	1.5421	5.2333	0.5040	6.7000	2.3511	**4.9000**	**0.4807**
Pendigit	12.1333	**0.3457**	14.0000	1.1142	10.7333	1.0483	12.6333	0.6687	**8.1333**	2.4598	13.2000	1.0635
Protein	9.4667	1.9429	13.6667	2.6566	10.2333	1.6750	9.7000	2.1520	8.6667	2.5506	**8.0000**	**1.3131**
Satelite	20.2000	**2.2500**	27.9670	3.1020	19.9330	2.4490	20.1330	2.3450	**16.8670**	3.7940	21.6000	2.3720
Spambase	32.7000	4.5040	38.5000	7.3426	28.5333	**2.3302**	30.8667	3.6078	**22.3667**	5.0068	26.7000	2.5482
Dermatology	11.9333	2.3034	24.1000	3.6041	17.2000	3.3672	11.5667	**1.5013**	14.4667	2.8495	**9.4333**	**1.5013**
Im	137.6000	18.5595	184.5333	37.0533	134.9667	8.0750	134.0000	8.7533	**110.4333**	11.9761	127.4667	**6.6162**
Vehicle	9.6000	**1.3287**	10.6000	2.7990	9.5000	2.0636	8.0333	**1.3257**	**7.4667**	1.9954	8.5333	1.9780
Average Features	19.4573	3.2047	29.4067	5.4231	23.0000	2.5449	21.2493	2.0259	18.5800	3.6083	19.0627	2.2052
Sum of the ranks	78		148		97		72		50		68	
Average rank	3.12		5.92		3.88		2.88		2		2.72	
Final rank	4		6		5		3		1		2	

posed BGOA-M; as it shows the lowest standard deviation values in the majority of the datasets.

Inspecting the results in terms of the number of selected features in Table 8, we observe that BBA performs better than other algorithms, followed by BGOA-M, GA and PSO in the second place with very competitive results. The difference between the aver-

age number of selected features by the three algorithms is very small for most of datasets. However, it is worth noting that for PenglungEW dataset which is highest dimensional dataset of 325 features, BGOA-M achieved the best classification accuracy of 93.4% with smallest number of features which are only 86 features on average. The very next method is GA with 88.9% accuracy with

Table 9

Comparison between BGOA-M and state-of-art methods based on average fitness values (AvgF).

Benchmark	BGOA_M		bGWO		BGSA		BPSO		BBA		GA	
	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev	AvgF	StdDev
Breastcancer	**0.0310**	**0.0000**	0.0451	0.0029	0.0470	0.0046	0.0412	**0.0000**	0.0388	0.0067	0.0341	0.0027
BreastEW	**0.0342**	0.0037	0.0714	0.0077	0.0645	0.0064	0.0383	**0.0035**	0.0662	0.0058	0.0386	0.0060
Exactly	**0.0046**	**0.0000**	0.2085	0.0676	0.2869	0.0853	**0.0046**	**0.0000**	0.2979	0.0910	**0.0046**	**0.0000**
Exactly2	0.2690	0.0020	0.2824	0.0235	0.2572	0.0277	**0.2305**	**0.0000**	0.2516	0.0104	0.2440	0.0166
HeartEW	0.1666	**0.0031**	0.2121	0.0151	0.2128	0.0209	0.1773	0.0108	0.1946	0.0243	**0.1584**	0.0136
Lymphography	**0.0924**	**0.0111**	0.1886	0.0267	0.1945	0.0224	0.1250	0.0123	0.2251	0.0290	0.1790	0.0151
M-of-n	**0.0046**	**0.0000**	0.1199	0.0498	0.1960	0.0532	**0.0046**	**0.0000**	0.1527	0.0613	**0.0046**	**0.0000**
penglungEW	**0.0678**	0.0149	0.1525	0.0204	0.2057	0.0191	0.1221	0.0165	0.1875	0.0198	0.1128	0.0202
SonarEW	0.0889	0.0105	0.1124	0.0140	0.1116	0.0114	0.0808	**0.0085**	0.1843	0.0183	**0.0683**	0.0191
SpectEW	0.1767	**0.0074**	0.1749	0.0124	0.1807	0.0118	**0.1301**	0.0110	0.1744	0.0205	0.1314	0.0139
CongressEW	**0.0264**	**0.0013**	0.0755	0.0092	0.0635	0.0096	0.0407	0.0030	0.0670	0.0156	0.0440	0.0079
IonosphereEW	**0.0570**	0.0074	0.1308	0.0140	0.0949	0.0143	0.0717	**0.0073**	0.1167	0.0123	0.0859	0.0172
KrvskpEW	0.0311	0.0030	0.0803	0.0123	0.1045	0.0474	0.0357	**0.0020**	0.1009	0.0336	**0.0243**	0.0038
Tic-tac-toe	0.2145	**0.0000**	0.2235	0.0171	0.2683	0.0210	**0.2056**	0.0009	0.2583	0.0249	0.2116	0.0076
Vote	**0.0393**	0.0024	0.0920	0.0085	0.0716	0.0091	0.0530	0.0043	0.0629	0.0116	0.0460	0.0113
WaveformEW	0.2516	0.0067	0.2765	0.0083	0.3062	0.0136	0.2531	**0.0041**	0.3046	0.0177	**0.2334**	0.0066
WineEW	**0.0145**	**0.0014**	0.0482	0.0108	0.0442	0.0102	0.0321	0.0053	0.0536	0.0154	0.0224	0.0063
Zoo	0.0458	0.0071	0.0521	0.0224	0.0441	0.0181	0.0427	0.0036	0.1156	0.0149	0.1163	0.0238
Pendigit	0.0119	0.0003	0.0165	0.0012	0.0187	0.0042	0.0106	0.0027	0.0230	0.0060	0.0135	0.0009
Protein	**0.1195**	0.0302	0.1842	0.0405	0.2031	0.0436	0.1685	**0.0198**	0.3173	0.0396	0.1360	0.0251
Satelite	0.0927	**0.0023**	0.0906	0.0032	0.0913	0.0037	0.0891	**0.0018**	0.1043	0.0049	**0.0821**	0.0039
Spambase	**0.0653**	0.0038	0.0867	0.0089	0.0935	0.0090	0.0766	**0.0007**	0.1104	0.0111	0.0694	0.0087
Dermatology	0.0035	0.0007	0.0356	0.0117	0.0430	0.0139	**0.0034**	**0.0004**	0.0420	0.0174	0.0050	0.0051
Im	**0.1590**	**0.0082**	0.2446	0.0129	0.2274	0.0106	0.2021	0.0092	0.2239	0.0129	0.1616	0.0161
Vehicle	**0.2327**	**0.0111**	0.3122	0.0212	0.278	0.0187	0.2766	**0.0111**	0.2909	0.018	0.2687	0.0154
Average Fitness	0.0920	0.0055	0.1407	0.0177	0.1484	0.0204	0.1006	0.0056	0.1586	0.0217	0.0998	0.0107
Sum of the ranks	48		117		123		54		124		53	
Average rank	1.92		4.68		4.92		2.16		4.96		2.12	
Final rank	1		4		5		2		6		3	

around 99 selected features. In terms of fitness values, the averages and their standard deviations are given in Table 9. Since the fitness function combines the classification accuracy with the number of selected features, from the results in Table 9, we can conclude that BGOA-M has the best performance among other approaches; since it has the best fitness values in 55% of the datasets. Moreover, the superiority of BGOA-M is proven when analyzing the standard deviation values, where it has the smallest values in 60% of the datasets.

The average convergence curves of all methods are drawn in Figs. 8 and 9, for all of the datasets, respectively. The figures show that BGOA-M is the fastest to converge when classifying the majority of the datasets. The best results of the classification accuracy, fitness values, and number of selected features are shown in Table 10. The results show that BGOA-M is best in reaching the highest classification accuracies in 50% of the datasets. In terms of fitness values, GA and BGOA-M were very competitive and showed the best fitness values in ten and nine datasets, respectively. In terms of the number of selected features, BBA performed the best. However, the difference between the results of BBA and BGOA-M is between two to five features in most cases. Moreover, for the high dimensional penglungEW dataset, BGOA-M achieved superior results with 97.3% accuracy and noticeable reduction in the number of selected features with only 36 features. While BBA achieved classification accuracy of 0.838 with 76 features.

The nonparametric Wilcoxons rank sum test is conducted at 5% significance level to verify whether there is a statistical difference between the obtained accuracy results of BGOA-M and each of the comparative methods. Table 11 shows the p-values of this test for BGOA-M against each method. In terms of classification accuracy, we observe that BGOA-M outperformed bGWO in 24 datasets, where in 23 out of these 24 datasets there is a significant difference (p-value less than 0.05). Comparing BGOA-M to BGSA, BPSO and GA, we can notice that BGOA-M has significantly outperformed them in 21, 16, and 14 datasets, respectively. Finally,

BGOA-M showed statistically remarkable classification accuracies compared to BBA in all datasets.

4.3. Comparisons with metaheuristics in the literature

In this section, we are interested in comparing the classification accuracy obtained from the proposed BGOA-M approach with some of the results reported in the literature. Table 12 contains the results of comparison between the average classification rates of BGOA-M and the results of GA and PSO algorithms that were reported in Kashef and Nezamabadi-pour (2015), and results of the bGWO1, bGWO2, GA, and PSO algorithms that are obtained from Emary et al. (2016b). Note that the results of SSA, ACO, GA1 and PSO1 were obtained by running the implementation of the authors in (López et al., 2006), and Kashef and Nezamabadi-pour (2015) with the same parameter settings. While the results of bGWO1, bGWO2, GA2, and PSO2 were obtained from Emary et al. (2016b), where the same datasets were used. In this section, we used only 18 datasets since only those datasets were used in Emary et al. (2016b).

Looking at the results in Table 12, it can be seen that BGOA-M has a substantial superiority among other approaches in obtaining the highest classification accuracy since it obtained the best results in 77% of the datasets. Comparing with the approaches of Kashef and Nezamabadi-pour (2015), BGOA-M shows a high performance in enhancing the classification accuracy since it outperformed GA1 in all of the datasets, and PSO1 in 17 out of 18 datasets. When BGOA-M is compared with the results of Emary et al. (2016b), it has also shown a superior performance; as it outperformed bGWO1 in 15 datasets, bGWO2 and GA2 in 16 datasets, and finally outperformed PSO2 in all of the datasets except WaveformEW dataset. When compared with ACO and SSA, it can be observed that it outperformed SSA and ACO on 17 and 14 datasets respectively. It is also worth mentioning that BGOA-M is not only outperforming other approaches, rather, there is a big dif-

Table 10
The best results of BGOA-M and state-of-art methods in terms of fitness values (F), classification accuracy (Acc) and number of selected features (NF).

Benchmark	BGOA-M			bGWO			BGSA			BPSO			BBA			GA		
	F	Acc	NF	F	Acc	NF	F	Acc	NF	F	Acc	NF	F	Acc	NF	F	Acc	NF
Breastcancer	0.0310	**0.9743**	5	0.0406	0.9657	3	0.0389	0.9686	4	0.0412	0.9629	4	**0.0271**	0.9714	**2**	0.0321	**0.9743**	4
BreastEW	0.0286	**0.9754**	10	0.0536	0.9509	11	0.0505	0.9544	10	0.0318	0.9719	9	0.0543	0.9474	**8**	**0.0280**	**0.9754**	**8**
Exactly	**0.0046**	**1.0000**	6	0.0390	0.9660	7	0.0113	0.9940	4	0.0046	1.0000	6	0.0418	0.6760	**3**	**0.0046**	**1.0000**	6
Exactly2	0.2683	0.7360	7	0.2411	0.7580	2	**0.2285**	**0.7700**	**1**	0.2305	0.7680	1	0.2463	0.7520	**1**	0.2305	**0.7700**	**1**
HeartEW	0.1652	0.8370	5	0.1806	0.8222	6	0.1741	0.8296	**3**	0.1636	0.8370	3	**0.1139**	0.8444	**3**	0.1285	**0.8741**	5
Lymphography	**0.0847**	**0.9189**	7	0.1388	0.8649	5	0.1388	0.8649	6	0.1124	0.8919	5	0.1644	0.8514	**3**	0.1639	0.8378	4
M-of-n	**0.0046**	**1.0000**	6	0.0161	0.9900	5	0.0113	0.9940	5	0.0046	1.0000	6	**0.0054**	**1.0000**	**2**	**0.0046**	**1.0000**	6
penglungEW	**0.0287**	**0.9730**	**36**	0.1120	0.8919	141	0.1698	0.8333	140	0.0848	0.9189	130	0.1629	0.8378	76	0.0832	0.9189	84
SonarEW	0.0716	0.9327	16	0.0743	0.9327	26	0.0897	0.9135	24	0.0619	0.9423	22	0.1383	0.8654	**15**	**0.0144**	**0.9904**	19
SpectEW	0.1602	0.8433	7	0.1398	0.8657	8	0.1579	0.8433	5	0.1158	0.8881	6	0.1357	0.8508	**4**	**0.1098**	**0.8955**	5
CongressEW	**0.0252**	**0.9771**	3	0.0582	0.9450	4	0.0388	0.9633	4	0.0382	0.9633	3	0.0452	0.9450	**2**	0.0362	0.9679	**2**
IonosphereEW	**0.0364**	**0.9659**	7	0.1045	0.8977	11	0.0624	0.9432	9	0.0539	0.9489	7	0.0947	0.9091	**6**	0.0527	0.9489	7
KrvskpEW	0.0259	0.9800	11	0.0582	0.9468	16	0.0513	0.9549	14	0.0325	0.9731	12	0.0561	0.9355	**11**	**0.0207**	**0.9850**	**11**
Tic-tac-toe	0.2145	0.7912	7	**0.1927**	**0.8121**	5	0.2278	0.7766	4	0.2051	0.7996	6	0.2102	0.7620	**2**	0.2062	0.7996	5
Vote	0.0368	0.9667	**3**	0.0698	0.9333	4	0.0434	0.9600	4	0.0471	0.9600	3	0.0421	0.9600	**3**	**0.0314**	**0.9733**	5
WaveformEW	0.2415	0.7604	14	0.2564	0.7476	25	0.2679	0.7344	14	0.2466	0.7560	15	0.2722	0.7228	**11**	**0.2187**	**0.7836**	15
WineEW	**0.0134**	**0.9888**	3	0.0299	0.9775	4	0.0261	0.9775	4	0.0276	0.9775	5	0.0253	0.9775	**2**	0.0165	**0.9888**	4
Zoo	0.0426	0.9608	5	**0.0244**	**0.9804**	8	**0.0238**	**0.9804**	6	0.0419	0.9608	5	0.1002	0.9020	**3**	0.1002	0.9020	4
Pendigit	0.0115	0.9960	12	0.0147	0.9933	11	0.0135	0.9940	9	**0.0095**	**0.9980**	11	0.0148	0.9907	**2**	0.0127	0.9960	10
Protein	**0.0511**	**0.9524**	6	0.1298	0.8750	8	0.1278	0.8750	8	0.1288	0.8750	4	0.2520	0.7500	**4**	0.0860	0.9167	5
Satelite	0.0881	0.9161	16	0.0839	0.9223	21	0.0818	0.9239	14	0.0835	0.9215	15	0.0947	0.9052	**9**	**0.0753**	**0.9301**	17
Spambase	0.0566	0.9501	26	0.0710	0.9349	26	0.0739	0.9305	24	0.0712	0.9327	26	0.0906	0.9153	**9**	**0.0524**	**0.9522**	21
Dermatology	0.0024	1.0000	8	0.0198	0.9865	17	0.0196	0.9865	11	0.0024	1.0000	8	0.0160	0.9865	9	**0.0021**	**1.0000**	**7**
Im	0.1448	0.8600	100	0.2085	0.7950	113	0.2079	0.7950	119	0.1780	0.8250	119	0.2014	0.7900	**86**	**0.1332**	**0.8700**	117
Vehicle	**0.2210**	**0.7824**	7	0.2629	0.7412	7	0.2379	0.7647	5	0.2607	0.7412	6	0.2496	0.7529	**4**	0.2432	0.7588	5
Average	0.0824	0.9215	13.3200	0.1048	0.8999	19.7600	0.1030	0.9010	18.0400	0.0911	0.9125	17.4800	0.1142	0.8720	11.2000	0.0835	0.9204	15.0800
Sum of the ranks	60	53	88	116	102	119	102	94	85	74	68	80	115	120	29	48	42	68
Average rank	2.4	2.12	3.52	4.64	4.08	4.76	4.08	3.76	3.4	2.96	2.72	3.2	4.6	4.8	1.16	1.92	1.68	2.72
Final rank	2	2	5	6	5	6	4	4	4	3	3	3	5	6	1	1	1	2

Table 11
P-values of the Wilcoxon test of BGOA-M classification accuracy results *vs* other algorithms (p $\geq$ 0.05 are underlined).

Dataset	bGWO	BGSA	BPSO	BBA	GA
Breast Cancer	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
BreastEW	< 0.05	< 0.05	< 0.05	< 0.05	0.004
Exactly	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Exactly2	< 0.05	0.216	< 0.05	< 0.05	< 0.05
HeartEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Lymphography	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
M-of-n	< 0.05	< 0.05	< 0.05	< 0.052	< 0.05
penglungEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
SonarEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
SpectEW	0.123	0.083	< 0.05	< 0.05	< 0.05
CongressEW	< 0.05	< 0.052	< 0.05	< 0.05	< 0.05
IonosphereEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
KrvskpEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Tic-tac-toe	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Vote	< 0.05	< 0.05	< 0.05	< 0.05	0.067
WaveformEW	< 0.05	< 0.05	0.160	< 0.05	< 0.05
WineEW	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Zoo	0.787	0.230	0.091	< 0.05	< 0.05
Pendigits	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Protein	< 0.05	< 0.05	< 0.05	< 0.05	0.083
Satellite	< 0.05	0.285	< 0.05	< 0.05	< 0.05
Spambase	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Dermatology	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
Im	< 0.05	< 0.05	< 0.05	< 0.05	0.737
Vehicle	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

ference between its results and the results of the other approaches. For example, in Exactly dataset, BGOA-M obtained 100% accuracy, then PSO1 comes next with a 97% accuracy, while in the third place comes the GA1 with a 82% accuracy. when comparing BGOA-M with bGWO1, bGWO2, GA2, and PSO2, the difference of classification accuracy started from 23%. The same observation can be made when inspecting the results of PenglungEW dataset where the result of BGOA-M is better than PSO1 by 6%, and GA1 comes in the second place with a difference of 26%. Note that PenglungEW is the largest dataset with 325 features. For other datasets, BGOA-M is still performing much better than other approaches.

4.4. Comparison with the filter-based methods

In this subsection, we compare the performance of BGOA-M (the best algorithm in the preceding section) with six filter-based methods which are:

- Information gain(IG) (Cover & Thomas, 2012)
- Spectrum (Zhao & Liu, 2007)
- Fisher Score (F-Score) (Duda, Hart & Stork, 2012)
- ReliefF (Kira and Rendell (1992) and Kononenko, Šimec and Robnik-Šikonja (1997))
- Correlation-based FS (CFS) (Hall & Smith, 1999)
- Fast Correlation Based Filter (FCBF) (Yu & Liu, 2003)

The first four algorithms are univariate filter-based methods (*i.e.,* IG, Spectrum, F-score, ReliefF), and they do not incorporate the dependencies between the input features in their evaluation criteria. In contrast to univariate methods, the last two algorithms (*i.e.* CFS and FCBF) are multivariate filter-based methods where the dependencies between the input features are taken into consideration for evaluation. These six algorithms are commonly used in the literature as standard FS methods, and are able to verify the results of the proposed techniques (Zhao, Morstatter, Sharma, Alelyani, Anand & Liu, 2010). All filters were adopted from the feature selection repository at Arizona State University (Zhao, Morstatter, Sharma, Alelyani, Anand & Liu, 2016). Several experiments were conducted using several number of features, where the top 5, 10, and 20 features were used to produce the classification accuracy. For most of the datasets, it has been found that using the top ranked 5 features performed well.

The results of all filter-based methods compared to the BGOA-M in terms of classification accuracy are shown in Table 13. Note that the results of BGOA-M are the average of 30 independent runs. It can be clearly seen that the BGOA-M has the best accuracy results in all datasets except for Breastcancer, KrvskpEW, and Vote datasets. The results of this experiment strongly support the findings of the literature which states that the wrapper-based methods tend to outperform the filter-based methods; because they incorporate the classification algorithm in evaluating the selected subset of features during the search process.

Table 12
Classification accuracies of the BGOA *versus* other meta-heuristics from the specialized literature.

	BGOA_M	SSA (*)	ACO (*)	GA1 (*)	PSO1 (*)	bGWO1 (*)	bGWO2 (*)	GA2 (*)	PSO2 (*)
Breastcancer	0.974	0.953	0.951	0.957	0.949	**0.976**	0.975	0.968	0.967
BreastEW	**0.970**	0.946	0.947	0.923	0.933	0.924	0.935	0.939	0.933
Exactly	**1.000**	0.718	0.756	0.822	0.973	0.708	0.776	0.674	0.688
Exactly2	0.735	0.703	**0.840**	0.677	0.666	0.745	0.750	0.746	0.730
HeartEW	**0.836**	0.777	0.801	0.732	0.745	0.776	0.776	0.780	0.787
Lymphography	**0.912**	0.781	0.859	0.758	0.759	0.744	0.700	0.696	0.744
M-of-n	**1.000**	0.819	0.853	0.916	0.996	0.908	0.963	0.861	0.921
penglungEW	**0.934**	0.872	0.845	0.672	0.879	0.600	0.584	0.584	0.584
SonarEW	**0.915**	0.835	0.830	0.833	0.804	0.731	0.729	0.754	0.737
SpectEW	0.826	0.778	**0.869**	0.756	0.738	0.820	0.822	0.793	0.822
CongressEW	**0.976**	0.940	0.919	0.898	0.937	0.935	0.938	0.932	0.928
IonosphereEW	**0.946**	0.834	0.853	0.863	0.876	0.807	0.834	0.814	0.819
KrvskpEW	**0.974**	0.896	0.901	0.940	0.949	0.944	0.956	0.920	0.941
Tic-tac-toe	0.791	0.750	**0.813**	0.764	0.750	0.728	0.727	0.719	0.735
Vote	**0.963**	0.937	0.926	0.808	0.888	0.912	0.920	0.904	0.904
WaveformEW	0.751	0.688	**0.813**	0.712	0.732	0.786	0.789	0.773	0.762
WineEW	**0.989**	0.951	0.944	0.947	0.937	0.930	0.920	0.937	0.933
Zoo	0.958	**0.980**	0.928	0.946	0.963	0.879	0.879	0.855	0.861
Average Accuracy	0.9139	0.8421	0.8693	0.8291	0.8597	0.8252	0.8318	0.8138	0.8220
Sum of the ranks	33	86	71	106	91	104	88	113	106
Average ranks	1.32	3.44	2.84	4.24	3.64	4.16	5.52	4.52	4.24
Final rank	1	3	2	7	5	6	4	9	7

* SSA (López et al., 2006); ACO (Kashef & Nezamabadi-pour, 2015); GA1 (Kashef & Nezamabadi-pour, 2015); PSO1 (Kashef & Nezamabadi-pour, 2015); bGWO1 (Emary et al., 2016b); bGWO2 (Emary et al., 2016b); GA2 (Emary et al., 2016b); PSO2 (Emary et al., 2016b)

Table 13
Classification accuracy results of all filter-based methods compared to the BGOA-M.

Dataset	CFS	FCBF	F-Score	IG	Spectrum	ReliefF	BGOA-M
Breastcancer	0.9571	**0.9857**	0.9786	0.9571	0.9571	0.9714	0.9743
BreastEW	0.8246	0.7982	0.9298	0.9298	0.7719	0.8158	**0.9697**
CongressEW	0.7931	0.7931	0.9080	0.8276	0.8276	0.8736	**1.0000**
Exactly	0.6700	0.4400	0.6000	0.6150	0.5750	0.6600	**0.7352**
Exactly2	0.7050	0.5450	0.6800	0.6200	0.6600	0.6900	**0.8358**
HeartEW	0.6481	0.6481	0.7593	0.7593	0.7963	0.8519	**0.9118**
IonosphereEW	0.8571	0.8571	0.7286	0.8000	0.8286	0.8429	**1.0000**
KrvskpEW	0.7684	0.9343	**0.9593**	0.9343	0.3772	0.9546	0.9342
Lymphography	0.5000	0.5667	0.6667	0.6667	0.7667	0.7000	**0.9147**
M-of-n	0.7850	0.8150	0.8150	0.8150	0.5800	**0.8300**	0.8261
PenglungEW	0.6000	0.6667	0.8000	0.6667	0.4000	0.4000	**0.9764**
SonarEW	0.3095	0.2143	0.0476	0.1905	0.0476	0.5238	**0.9458**
SpectEW	0.7358	0.7736	0.7925	0.7925	0.7358	0.8491	**0.9736**
Tic-tac-toe	0.0000	0.0000	0.0104	0.0104	0.1667	0.0938	**0.7912**
Vote	0.9500	0.9500	0.9333	**0.9667**	0.8500	0.9000	0.9633
WaveformEW	0.6200	0.7100	0.6620	0.6620	0.2920	0.7420	**0.7511**
WineEW	0.7778	0.8889	0.8611	0.8889	0.8889	0.9444	**0.9888**
Zoo	0.8000	0.9000	0.6500	0.8500	0.6000	0.8000	**0.9575**
Pendigit	0.8566	0.8826	0.9006	0.8506	0.8472	0.8906	**0.9956**
Protein	0.3913	0.4783	0.4783	0.5217	0.1739	0.4783	**0.8841**
Satelite	0.8190	0.8718	0.8089	0.8174	0.8081	0.5245	**0.9120**
Spambase	0.3420	0.3793	0.5283	0.5685	0.7228	0.3707	**0.9398**
Dermatology	0.7534	0.9315	0.2740	0.7260	0.3836	0.8493	**1.0000**
Im	0.5450	0.5750	0.4850	0.5350	0.2650	0.5100	**0.8445**
Vehicle	0.6805	0.5621	0.6095	0.6095	0.5207	0.5917	**0.7704**
Average Accuracy	0.6676	0.6867	0.6747	0.7032	0.5937	0.7063	0.9118
Sum of the ranks	115	99	100	94	136	88	33
Average rank	4.6	3.96	4.00	3.76	5.44	3.52	1.32
Final rank	6	4	5	3	7	2	1

Taken together, the results showed that the modifications incorporated to the GOA algorithm make this algorithm a very reliable and accurate alternative as compared to the current wrapper-based FS methods in the literature. The discrepancy of the results were more evident on challenging case studies, showing the merits of the proposed algorithms when solving high-dimensional problems. Referring to the NFL theorem, the proposed algorithms are not able to solve all FS problems. However, it is definitely worth to consider them as promising techniques with the potential of solving challenging FS problems.

Among the proposed algorithms, BGOA-M performed best. This shows that for some metaheuristics, transfer functions are not the best option. Transfer functions are easy to use, yet the results of this paper showed that other conversion mechanisms can be more effective in converting a continuous algorithm than a binary one. One of the reasons is that in a binary search space, exploration is not well-defined, since the maximum step size is always equal to 1. Therefore, the frequency of changing the bits will encourage more exploration. The proposed conversion mechanism benefits from this fact and assisted both BGOA and BGOA-M significantly to outperform other algorithms on the majority of case studies. This conclusion can be made when comparing the results of BGOA (that uses the proposed conversion mechanism), with the results of BGOA-S and BGOA-V, that use the S-shaped and V-shaped transfer functions, respectively. Moreover, the employed evolutionary operator i.e., mutation, has significantly affected the performance of BGOA approach by enhancing the exploration ability of GOA algorithm.

5. Conclusion

In this paper, two novel binary versions of the GOA algorithm were proposed and applied to FS problems. The first approach is based on transfer functions, while the second approach uses a novel mechanism that considers the position of the best solution so far to reposition the current solution. To benchmark the pro-

posed algorithms, 25 standard UCI benchmark datasets were used. The results were compared with 11 FS methods in the literature including five metaheuristics (wrapper-based) and six filter-based. The results demonstrated the superiority of the proposed methods. It was observed that the mutation operators highly promote the exploration in FS problems; since the binary bits are changed more frequently. Quantitative and qualitative results also indicated that the convergence speed of the proposed algorithm is high, which resulted in finding an accurate set of features for the majority of the case studies. According to the results, discussions, and analyses, it is concluded that the proposed binary GOA (specially BGOA-M) have merits among the current FS algorithms and worthy of consideration when solving challenging FS problems.

For future work, the application of the proposed approaches to other real word problems can be investigated such as real business problems, and medical applications. Furthermore, the performance of the proposed approaches can be investigated using other classifiers as evaluators such as extreme machine learning, neural networks, and support vector machines.

References

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications, 36*(3), 6843–6853.

Aladeemy, M., Tutun, S., & Khasawneh, M. T. (2017). A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. *Expert Systems with Applications, 88*, 118–131.

Alsaafin, A., & Elnagar, A. (2017). A minimal subset of features using feature selection for handwritten digit recognition. *Journal of Intelligent Learning Systems and Applications, 9*(04), 55.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.

Asir, D., Appavu, S., & Jebamalar, E. (2016). Literature review on feature selection methods for high-dimensional data. *International Journal of Computer Applications, 136*(1), 9–17.

Baig, M. Z., Aslam, N., Shum, H. P., & Zhang, L. (2017). Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery eeg. *Expert Systems with Applications, 90*, 184–195.

Chakraborty, B. (2008). Feature subset selection by particle swarm optimization with fuzzy fitness function. In *Proceedings of the 3rd international con-*

ference on intelligent system and knowledge engineering, 2008. ISKE 2008.: 1 (pp. 1038–1042). IEEE.

Chen, H., Jiang, W., Li, C., & Li, R. (2013). A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. *Mathematical problems in Engineering, 2013*, 6 Article ID 524017. doi:10.1155/2013/524017.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Dadaneh, B. Z., Markid, H. Y., & Zakerolhosseini, A. (2016). Unsupervised probabilistic feature selection using ant colony optimization. *Expert Systems with Applications, 53*, 27–42.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(1–4), 131–156.

Dua, D., & Karra Taniskidou, E. (2017). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences http://archive.ics.uci.edu/ml.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016a). Binary ant lion approaches for feature selection. *Neurocomputing, 213*, 54–65.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016b). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381.

Ghareb, A. S., Bakar, A. A., & Hamdan, A. R. (2016). Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Systems with Applications, 49*, 31–47.

Glover, F. W., & Kochenberger, G. A. (2006). *Handbook of metaheuristics*: 57. Springer Science & Business Media.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(Mar), 1157–1182.

Hall, M. A., & Smith, L. A. (1999). Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper.. In *Proceedings of the FLAIRS conference: 1999* (pp. 235–239).

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Elsevier.

Hedar, A.-R., Wang, J., & Fukushima, M. (2008). Tabu search for attribute reduction in rough set theory. *Soft Computing-A Fusion of Foundations, Methodologies and Applications, 12*(9), 909–918.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. *IEEE Transactions on Knowledge and Data Engineering, 16*(12), 1457–1471.

Kabir, M. M., Shahjahan, M., & Murase, K. (2009). An efficient feature selection using ant colony optimization algorithm. In *Proceedings of the international conference on neural information processing* (pp. 242–252). Springer.

Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing, 74*(17), 2914–2928.

Kabir, M. M., Shahjahan, M., & Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications, 39*(3), 3747–3763.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report*. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing, 147*, 271–279.

Katrutsa, A., & Strijov, V. (2017). Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications, 76*, 1–11.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Proceedings of the 1997 IEEE international conference on systems, man, and cybernetics, 1997. Computational cybernetics and simulation.: 5* (pp. 4104–4108). IEEE.

Kennedy, R. (1995). J. and eberhart, particle swarm optimization. In *Proceedings of IEEE international conference on neural networks IV, pages: 1000*.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Aaai: 2* (pp. 129–134).

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence, 97*(1–2), 273–324.

Kononenko, I., Šimec, E., & Robnik-Šikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence, 7*(1), 39–55.

Kulkarni, A. J., Durugkar, I. P., & Kumar, M. (2013). Cohort intelligence: a self supervised learning behavior. In *Proceedings of the 2013 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 1396–1400). IEEE.

Li, P., Shrivastava, A., Moore, J. L., & König, A. C. (2011). Hashing algorithms for large-scale learning. In *Advances in neural information processing systems* (pp. 2672–2680).

Li, Y., Li, T., & Liu, H. (2017). Recent advances in feature selection and its applications. *Knowledge and Information Systems, 53*(3), 551–577.

Liu, H., & Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective*: 453. Springer Science & Business Media.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering, 17*(4), 491–502.

López, F. C. G., Torres, M. G., Pérez, J. A. M., & Vega, J. M. M. (2004). Scatter search for the feature selection problem. In *Current topics in artificial intelligence* (pp. 517–525). Springer.

López, F. G., Torres, M. G., Batista, B. M., Pérez, J. A. M., & Moreno-Vega, J. M. (2006). Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research, 169*(2), 477–489.

Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*.

Mirjalili, S., & Lewis, A. (2013). S-Shaped *versus* v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1–14.

Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing, 43*, 117–130.

Oh, I.-S., Lee, J.-S., & Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(11), 1424–1437.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). Bgsa: binary gravitational search algorithm. *Natural Computing, 9*(3), 727–745.

Rogers, S. M., Matheson, T., Despland, E., Dodgson, T., Burrows, M., & Simpson, S. J. (2003). Mechanosensory-induced behavioural gregarization in the desert locust schistocerca gregaria. *Journal of Experimental Biology, 206*(22), 3991–4002.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47.

Shokouhifar, M., & Sabet, S. (2010). A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization. In *Proceedings of the 3rd international conference on machine vision (ICMV 2010)* (pp. 502–506).

Simpson, S. J., McCAFFERY, A. R., & HAeGELE, B. F. (1999). A behavioural analysis of phase change in the desert locust. *Biological Reviews, 74*(4), 461–480.

Song, Q., Jiang, H., & Liu, J. (2017). Feature selection based on fda and f-score for multi-class classification. *Expert Systems with Applications, 81*, 22–27.

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*: 74. John Wiley & Sons.

Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimization algorithm. *Applied Soft Computing, 49*, 248–258.

Wang, J., Ci, L.-l., & YAO, K.-z. (2005). A survey of feature selection. *Computer Engineering & Science, 12*, 023.

Wang, J., Hedar, A.-R., Wang, S., & Ma, J. (2012). Rough set and scatter search metaheuristic based feature selection for credit scoring. *Expert Systems with Applications, 39*(6), 6123–6128.

Wang, J., Li, T., & Ren, R. (2010). A real time idss based on artificial bee colony–support vector machine algorithm. In *Proceedings of the 2010 third international workshop on advanced computational intelligence (IWACI)* (pp. 91–96). IEEE.

Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters, 28*(4), 459–471.

Wieland, M., & Pittore, M. (2014). Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images. *Remote Sensing, 6*(4), 2912–2939.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.

Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing, 18*, 261–276.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications, 13*(2), 44–49.

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 856–863).

Zarshenas, A., & Suzuki, K. (2016). Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning. *Knowledge-Based Systems, 110*, 191–201.

Zawbaa, H. M., Emary, E., & Grosan, C. (2016). Feature selection via chaotic antlion optimization. *PloS One, 11*(3), e0150652.

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 1151–1157). ACM.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., & Liu, H. (2010). Advancing feature selection research. *ASU Feature Selection Repository*, 1–28.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H., 2016. Advancing feature selection research-asu feature selection repository. Arizona State University. 2016.

Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications, 62*, 91–103.