An Evolutionary Gravitational Search-based Feature Selection

Article *in* Information Sciences · May 2019

DOI: 10.1016/j.ins.2019.05.038

CITATIONS
5

READS
711

7 authors, including:

Majdi Mafarja
Birzeit University
45 PUBLICATIONS **633** CITATIONS
SEE PROFILE

Ali Asghar Heidari
National University of Singapore
30 PUBLICATIONS **517** CITATIONS
SEE PROFILE

Hossam Faris
University of Jordan
134 PUBLICATIONS **1,567** CITATIONS
SEE PROFILE

Ibrahim Aljarah
University of Jordan
88 PUBLICATIONS **1,109** CITATIONS
SEE PROFILE

Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Sciences

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

An evolutionary gravitational search-based feature selection

Q1 Mohammad Taradeh[a], Majdi Mafarja[b,*], Ali Asghar Heidari[c,d], Hossam Faris[e], Ibrahim Aljarah[e], Seyedali Mirjalili[f], Hamido Fujita[g]

[a] Faculty of Engineering and Technology, Birzeit University, Birzeit, Palestine
[b] Department of Computer Science, Birzeit University, Birzeit, Palestine
[c] School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran
[d] Department of Computer Science, School of Computing, National University of Singapore, Singapore
[e] King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
[f] Institute of Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia
[g] Faculty of Software and Information Science, Iwate Prefectural University (IPU), Iwate, Japan

ARTICLE INFO

Article history:
Received 1 October 2018
Revised 22 February 2019
Accepted 12 May 2019
Available online xxx

Keywords:
Gravitational search algorithm
Genetic algorithm
Feature selection
Supervised learning
Classification
Optimization

ABSTRACT

With recent advancements in data collection tools and the widespread use of intelligent information systems, a huge amount of data streams with lots of redundant, irrelevant, and noisy features are collected and a large number of features (attributes) should be processed. Therefore, there is a growing demand for developing efficient Feature Selection (FS) techniques. Gravitational Search algorithm (GSA) is a successful population-based metaheuristic inspired by Newton's law of gravity. In this research, a novel GSA-based algorithm with evolutionary crossover and mutation operators is proposed to deal with feature selection (FS) tasks. As an NP-hard problem, FS finds an optimal subset of features from a given set. For the proposed wrapper FS method, both K-Nearest Neighbors (KNN) and Decision Tree (DT) classifiers are used as evaluators. Eighteen well-known UCI datasets are utilized to assess the performance of the proposed approaches. In order to verify the efficiency of proposed algorithms, the results are compared with some popular nature-inspired algorithms (i.e. Genetic Algorithm (GA), Particle Swarm Optimizer (PSO), and Grey Wolf Optimizer (GWO)). The extensive results and comparisons demonstrate the superiority of the proposed algorithm in solving FS problems.

1. Introduction

Data mining is an automatic or a semi-automatic process of extracting and discovering implicit, unknown, and potentially useful patterns and information from massive data stored and captured from date repositories (web, database, data warehousing) [15]. Data mining tasks are usually divided into two categories: predictive and descriptive. The objective of the predictive tasks is to predict or classify for determining which class an instance or example belongs, based on the values of some features (i.e, independent or conditional features) in a dataset. An example of the prediction task is to predict

* Corresponding author.
E-mail addresses: mohammad.taradeh@gmail.com (M. Taradeh), mmafarja@birzeit.edu (M. Mafarja), as_heidari@ut.ac.ir, aliasghar68@gmail.com, aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu (A.A. Heidari), hossam.faris@ju.edu.jo (H. Faris), i.aljarah@ju.edu.jo (I. Aljarah), seyedali.mirjalili@griffithuni.edu.au (S. Mirjalili), hfujita-799@acm.org (H. Fujita).

https://doi.org/10.1016/j.ins.2019.05.038
0020-0255/

if a new patient (instance) is in danger of a heart attack disease or not based on some clinical tests. Descriptive tasks aim to find clusters, correlations, and trends based on the implicit relationships hidden in the underlying data.

With advanced data collection tools and since the collected data is not specified for the data mining purposes, it may contain redundant or irrelevant features [28]. Applying data mining on such data might lead to misleading results. Thus, data pre-processing is an essential step to refine the data to be used in any learning model. The main purpose of the pre-processing step is to clean and transform raw data into a suitable format, which improves the performance of data mining tasks [15].

One of the most important pre-processing techniques is dimensionality reduction (DR), which aims to reduce the number of features in a dataset to the minimum and improve the performance of different data mining tasks (e.g., classification, clustering, association rule mining, etc.). One of the most essential tools in DR is Feature Selection (FS).

FS is the process of selecting the minimal representative feature subset from the original set of features to satisfy a measuring criterion. By eliminating the redundant and irrelevant features from a dataset, the performance of the consequent data mining tasks can be improved substantially in terms of the computational time and/or the learning performance (e.g., the classification accuracy of a classifier). The standard FS process has four major phases: subset generation, subset evaluation, stopping criteria, and validation.

Subsets generation is considered as a search problem that aims to select the best subset of all possible feature subsets. Then, the selected subset can be evaluated using statistical measures (i.e., the correlation between the feature and the target class) as in filter approaches, or by using a classifiers measure (e.g., accuracy) to evaluate the subset performance as in wrapper approaches. When employing an FS method for a dataset, the process is repeated until a stopping constraint is met (i.e., achieving a predefined goal (classification accuracy), or repeating the process for a certain number of iterations). The validation step is usually performed after the FS process to assess the quality of the resulted feature subset.

Searching for the best feature subset is a challenge in the FS process, which is considered as a combinatorial NP-hard problem [29]. Complete Search strategies (e.g. exhaustive) examine all possible solutions (feature subsets) for the search problem. Formally speaking, for a dataset of N features, the complete search tends to evaluate $2^N - 1$ possible subsets for selecting one of them. Therefore, it is impractical to use this strategy to tackle the FS problem for large values of N. Another strategy that could be used to combat this issue is random search strategy [2].

A random search can generate subsets constantly and keep improving the quality of the selected features in an iterative manner. In each step, the next subset is obtained randomly using the information collected in the past steps. In the worst case, a random search continues evaluating all possible solutions as the complete search. Heuristic searches belong to the third class of search methods can be used to find an optimal subset of features. Over the last decade, many researchers showed the effectiveness of using metaheuristic algorithms in solving FS problems [41]. This is due to the ability of those algorithms to find (near) optimal solutions in a reasonable time [9].

Meta-heuristics use heuristic information, but they are considered general-purpose optimization techniques that find reasonable solutions in a reasonable time for optimization problems. According to Talbi [47], metaheuristic algorithms can be classified into two main families: single-solution-based (S-Metaheuristics) and population-based (P-metaheuristics). The most popular examples of S-Metaheuristics are Simulated Annealing (SA) and Tabu Search (TS). Examples of P-metaheuristics are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). Some of the recent optimizers are Grey Wolf Optimizer (GWO), Salp Swarm Algorithm (SSA) and Harris Hawks Optimization (HHO) [19]. The aforementioned algorithms demonstrate superior efficiency in tackling FS problems when compared to the exact methods [30]. Another class of metaheuristic methods that can be added to the latter taxonomy is the Memetic Algorithms (MAs). This class can be broadly defined as hybrid algorithms that are composed of an evolutionary algorithm and one or more local search algorithms within the same generation cycle [3–5,21,39].

Swarm Intelligence (SI) algorithms are mostly P-metaheuristics, which search for the (near) optimal solution for a specific problem by employing a set of search agents to explore the search space [11]. The key characteristic of SI algorithms is that they try to mimic the natural behavior of some creatures that live in groups like folks of birds, ants, bees and others [1]. PSO, ACO, GWO, HHO, SSA, and FA are all examples of SI algorithms that mimic the social behavior of some creatures. Moreover, some SI algorithms have been inspired by other phenomena in nature, like physics-based algorithms that are based on physical laws. As an example, Gravitational Search Algorithm (GSA) [42] is a physics-based algorithm inspired by Newton's law of universal gravitation and Newton's law of motion.

Two common characteristics between all SI algorithms are exploration and exploitation that should be carefully considered in order to achieve the best performance in finding the (near) optimum solution. Exploration aims to explore the whole search space in order to find promising solutions in undiscovered areas. By contrast, the exploitation phase aims to improve the already discovered solutions by searching their neighborhood. Both phases are associated, and must efficiently cover the search space to search the (near) optimal solution; however, the key point is to find the right balance between them. Indeed, by having more exploitation than exploration, the algorithm will be stuck in locally optimal solutions. At the same time, the algorithm will lose some best solutions if it goes far in exploration.

In this paper, an efficient hybrid FS approach, which is called HGSA, is proposed to combine the benefits of the SI algorithms and the GA for improving the exploitative and exploitative capabilities of the GSA. The rest of the paper is organized as follows: Section 2 provides the literature review of optimization algorithms applied to FS problems. GSA algorithm is presented in Section 3. Section 4 covers the proposed method. The results and conclusions are given in Sections 5 and 6, respectively.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

2. Literature review

FS is one of the most important and frequently used techniques as a pre-processing tool for enhancing different data mining techniques. Therefore, many efforts have been put into improving the FS methods. In recent years, many researchers involved the SI algorithms (e.g., GA, GSA, PSO, and GWO) for solving FS problems.

The GA is an Evolutionary Algorithm (EA) that solves optimization problems using techniques motivated by natural evolution including inheritance, mutation, selection, and crossover. GA was used for the first time in solving the FS problems by Yang and Honavar [49]. As other P-metaheuristic algorithms, GA is more exploration-oriented than exploitation, accordingly, local search mechanisms were hybridized with GA algorithm to achieve better results, and this is considered as the first hybrid metaheuristic algorithm for FS in the literature [37]. In their approach, local search schemes were integrated with the GAs to fine-tune the searching process. The mechanisms are generated with regard to their fine-tuning capacity and the corresponding efficiency and computational times are investigated based on different FS datasets.

Eberhart and Kennedy [7] proposed a breakthrough computation technique known as PSO, which is inspired by the social intelligence of a flock of birds. Normally a leader leads the swarm of birds or fish, and every single particle of the swarm follows the leading navigator based on their intuition. Kennedy and Eberhart [24] proposed a binary version of the PSO (called BPSO), to solve the binary problems. BPSO was used for the first time in the FS domain by Firpi and Goodman [13]. BPSO has been widely used to tackle FS problems, where many researchers have adopted the BPSO for achieving better results in finding the best subset of features.

Recently, many optimization algorithms have been proposed and used to tackle FS problems. Grey Wolf Optimizer (GWO) is a newly proposed SI algorithm developed by Mirjalili et al. [33]. The GWO was used in many approaches for solving various problems [10]. The GWO mimics the hierarchical domination and hunting mechanism of Grey wolves in nature. A binary version of GWO was proposed and used to select optimal feature subset for classification in [8]. Li et al. [26] first proposed an improved GWO (IGWO) as a wrapper-based on kernel extreme machine learning (KELM) for FS tasks. They then used IGWO for finding the optimal feature subset for medical data. In the proposed approach, GA was first adapted to generate the diversified initial positions, and then, GWO was used to update the current positions of the population in a discrete searching space. Pathak et al. [38] proposed a levy flight-based GWO to deal with FS for image steganalysis. Whale Optimization Algorithm (WOA) [17] is a recent SI algorithm that mimics the foraging behavior of humpback whales. A binary version of WOA was proposed in [31] and was used to tackle FS problems. Mafarja and Mirjalili [32] proposed a hybrid approach that combined SA with WOA for the FS problem. The Grasshopper Optimization Algorithm (GOA) [18] is another recent population-based metaheuristic that mimics the swarming behaviors of grasshoppers, and was improved by hybridizing it with an evolutionary operator to be employed as searching strategy in a novel FS approach [30].

GSA is an optimization algorithm developed by Rashedi et al. [42]. Newton's law of gravity and motion drives the main mathematical modes of GSA. The population in GSA is presented as a group of physical particles each of which has a mass, which presents the solution's fitness value. The heavier mass, which is supposed to be the better solution, is moving slower than the lighter mass. This assumption controls the exploration and exploitation behaviors in GSA. A binary version of GSA (called BGSA) was developed by Rashedi et al. [43] for solving the binary problems. Although BGSA was employed to solve FS problems, according to [40], BGSA still suffers from two well-known issues; falling in the local optimum and the slow convergence rate. Rashedi and Nezamabadi-pour [41] proposed an improved version of the BGSA to overcome the stagnation situation and the slow convergence rate of the original GSA, by changing the transfer function to improve the exploration ability of the algorithm. They also used an elitism strategy to improve the movement of the particles toward the best solution.

An astrophysics-based disruption mechanism was added to GSA by Sarafrazi et al. [45]. To deal with hydraulic turbine systems, a PSO-based searching strategy was embedded in GSA by Li and Zhou [25]. Shaw et al. [46] developed an opposition-based learning-based (OBL) GSA to speed up the convergence behavior and generate some jumping mechanisms. In [50], two modifications were considered: to include the upper and lower limits of the d-th dimension and to propose a new gravitational constant. By this method, they boost the objects' diversity and brings more exploration to the basic method. In [6], an enhanced GSA based on PSO and elastic-ball strategy is designed. In [16], they integrated Piece Wise Linear (PWL) chaotic map and sequential quadratic programming into the basic GSA. The PWL map was combined with GSA to improve the exploration, and then they used sequential quadratic programming to improve the exploitation for the global best. In [34], ten chaotic maps were used to update the gravitational constant (G) of the GSA. In addition, an adaptive normalization method was proposed to transit from the exploration phase to the exploitation phase, more smoothly. In 2018, Rashedi et al. [44] provided a comprehensive survey on different variants and operators of GSA algorithm as well.

Based on literature review, it can be seen that several SI algorithms have been utilized to serve as search strategies in both filter and wrapper FS approaches. According to the no free lunch theorem (NFL) [48], there is no algorithm that can solve all optimization problems efficiently, and if an algorithm showed a superior performance in solving a specific problem, it may fail to find good solutions for another. The oscillating behavior of the optimizers can be interpreted to some characteristics in each algorithm. Moreover, the nature of the problem being solved has a vital impact on the performance of the algorithm. This motivated our attempts to propose a new FS approach that try to overcome the main drawbacks of the GSA algorithm. The low convergence rate of the GSA is treated by employing a logarithmic decreasing strategy to update the value of the gravitational constant (G). Moreover, the exploration and exploitation abilities of the GSA are enhanced by proposing a novel mechanism that works based on two evolutionary operators (i.e., crossover and mutation), where the

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

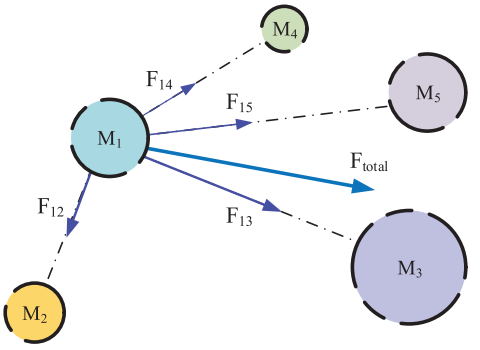

Fig. 1. Interaction of agents in GSA.

whole population is improved instead of improving some individuals of the population as in the traditional evolutionary algorithms (e.g., GA). To our knowledge, there is no such methods in literature to hybridize the GSA with the evolutionary operators.

3. Preliminaries

3.1. Gravitational search algorithm (GSA)

GSA [42] is a successful metaheuristic inspired by Newton's gravitational and motion laws. In GSA, search agents have the role of interacting physical objects, and the performance of the solutions is seen as the mass of the objects. The particles can attract other agents based on the gravitational force. This force pushes lighter objects towards the heavier objects. The heavier objects, which are considered as better solutions, will move slower than the lighter objects (see Fig. 1). This behavior ensures the exploitation phase in GSA. Consider N search agents (masses) in a search space with n dimensions, the position of the $i-th$ solution (mass) is defined as vector X_i as in Eq. (1):

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n), \quad i = 1, 2, \ldots, N \tag{1}$$

Where the value of each vector represents the position of agents in that dimension, for instance x_i^j is the position of the $i-th$ agent in the j dimension. As mentioned before, the mass M_i for $i-th$ agent is calculated based on the fitness of that agent at a specific time (iteration) t as in Eq. (2):

$$M_i(t) = \frac{fit_i(t) - worst(t)}{\sum_{j=1}^{N} (fit_j(t) - worst(t))} \tag{2}$$

where M_i and $fit_i(t)$ are the mass and the fitness of $i-th$ agent at time t, and $worst(t)$ is the worst fitness among all agents at time t (current population). In each iteration, agents change their position to discover the unexplored areas inside the feature space.

In GSA, the new position for the $i-th$ agent can be calculated by summing the current position of the agent with its next velocity $v_i(t+1)$ as in Eq. (4):

$$X_i(t+1) = X_i(t) + v_i(t+1) \tag{3}$$

where the next velocity of an object is the acceleration of the object added to its current velocity:

$$v_i(t+1) = r \times v_i(t) + a_i(t) \tag{4}$$

where r is a random variable between (0,1). According to the laws of motion, the acceleration of the $i-th$ particle can be calculated as the total gravitational force of other particles divided by the mass of particle i as in Eq. (5).

$$a_i(t) = \frac{\sum_{j=1, j\neq i}^{N} (q \times F_{ij}(t))}{M_i(t)} \tag{5}$$

where q is a random variable between (0,1), $F_{ij}(t)$ denotes the gravitational force acting by particle j on particle i at specific iteration t. According to the Newton's gravitational law, this force can be calculated as in Eq. (6).

$$F_{ij}(t) = G(t)\frac{M_i(t) \times M_j(t)}{R^2}(X_j(t) - X_i(t)) \tag{6}$$

where G is the gravitational constant. R is the distance between particle j and particle i. In GSA, it G is a decreasing function and the initial value $G(G_0, t)$ is used to control the search process. Pseudocode of GSA is described in Algorithm 1.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Algorithm 1 Pseudocode of GSA.

Initialize the candidate objects $X_i (i = 1, 2, \ldots, N)$
while (end condition is not met) **do**
  Evaluate the fitness of all objects
  Calculate *best*, *worst*, M_i, M_j
  Update the gravitational factor $G(t)$
  Calculate forces F_{ij} by Eq. (6).
  Update acceleration by Eq. (5)
  Update velocities using Eq. (4)
  Update positions by Eq. (3)
Return the best search agent

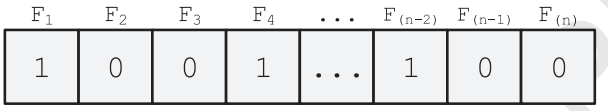

Fig. 2. Binary solution representation.

3.2. Learning algorithms

In this paper, the wrapper FS model is adopted to assess the goodness of each feature subset, in which two learning algorithms (i.e., KNN and DT) from different families are used in all experiments. The classifiers employed are described as follows:

3.2.1. KNN Classifier

KNN classifier is one of the simplest classification algorithms widely used in literature [37]. KNN tries to find the training nearest neighbors to the test sample according to Euclidean distance, and then the test sample is classified based on the majority class of its nearest neighbors. In KNNs, there is only one parameter to be determined: the number of neighbors (k) to be considered when classifying a new test sample. If we set k to one, the test sample is basically assigned to the class of that identical nearest neighbor. However, the value of k parameter has a major effect on the overall classification accuracy of the KNN classifier. Often, Euclidean distance is employed in KNN using Eq. (7):

$$ED(X_1, X_2) = \left(\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2 \right)^{\frac{1}{2}} \tag{7}$$

where X_1 and X_2 denote two points with n dimensions. The KNN is categorized as an instance-based learning model, which is one of the simplest algorithms among all methods in machine learning. In the course of the learning stage, there is no abstraction of training information in instance-based learners.

3.2.2. DT Classifier

DT is a simple and commonly used classifier in data mining [12]. The goal of DT is to create a model for predicting the value of the target feature (class) for a test sample based on the values of the input features vector. It follows the top-down schema, by choosing the feature at each level, that "best split" the training data into subsets based on the gain-ratio measure. The goal is to find the most homogeneous set based on the value of the target feature (class). Different decision tree construction algorithms use different homogeneity metrics. For example, Classification And Regression Tree (CART) uses Gini impurity and Variance reduction, while ID3 and C4.5 algorithms use Information gain.

3.3. GSA For feature selection

In general, FS methods aim to select the most informative features among the original set of features by eliminating the redundant and/or irrelevant features. Therefore, it is evident to use a binary representation for representing a solution of the FS problem. In a binary solution, each element indicates selecting or not selecting the corresponding feature in the original dataset. Therefore, each dimension in the solution takes either 1 (selected) or 0 (not selected) as shown in Fig. 2.

The GSA was originally designed to tackle the optimization problems with continuous search spaces, so to employ GSA as a search strategy within a FS method, which is a binary problem, the solutions must be converted to a binary form. In BGSA [43], a v-shaped transfer function (TF), which is shown in Eq. (8) (see Fig. 3), was utilized to convert the continuous solutions to binary where the velocity vector in Eq. (4) was used as input.

In this case, each element in the velocity vector represents the probability of flipping the corresponding feature from selected to not selected and vice versa; i.e., the dimensions with high velocity values have high probability to flip their

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

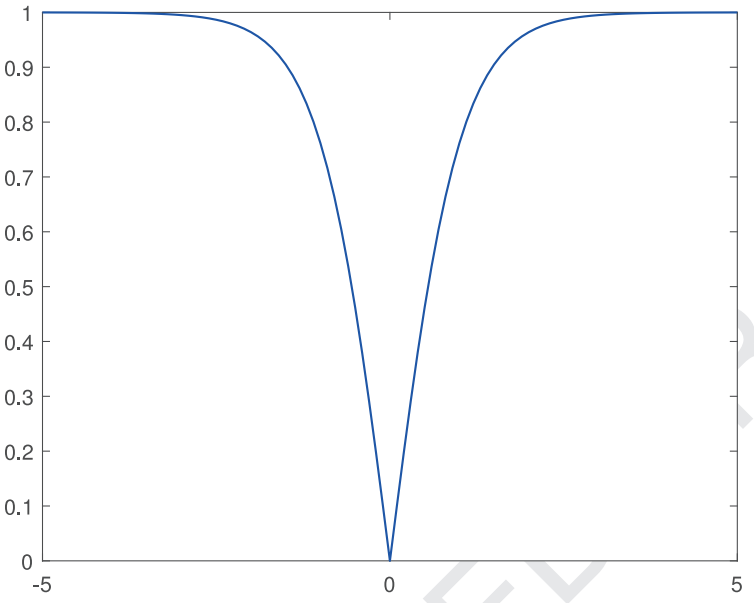

Fig. 3. Used transfer function.

values, while the dimensions with lower velocities have lower chances [36].

$$TF(v_d^i(t)) = |tanh(v_d^i(t))| \tag{8}$$

where $v_d^i(t)$ is the velocity value of the i^{th} element in d^{th} dimension in t^{th} iteration.

The result $TF(v_d^i(t))$, obtained from Eq. (8) is then used to convert a position's element to 0 or 1 according to Eq. (9):

$$X(t+1) = \begin{cases} -X_t & r < TF(v_k^i(t)) \\ X_t & r \geq TF(v_k^i(t)) \end{cases} \tag{9}$$

where r is a random number inside (0, 1).

As for any optimization problem, designing an appropriate objective function, which is a part of the problem formulation, is crucial in tackling the FS problem. In GSA, the physical mass of a solution represents the quality of that solution, i.e., the best solution is much heavier than the worst one. When considering the GSA for FS and since the solution represents the feature subset, the goodness of a solution can be considered based on two factors; the (minimal) number of features included in this subset, and the (maximal) classification accuracy that can be obtained by using the selected features. Therefore, an objective function that combines these two factors is used in this work (see Eq. (10)).

$$Fitness = min\left[\alpha\gamma_R(D) + \beta\frac{|R|}{|N|}\right] \tag{10}$$

where $\gamma_R(D)$ represents the classification error rate of the KNN classifier, $|R|$ is the number of selected features, $|N|$ is the number of original features in the dataset, α and β are the weighting factors of the classification error rate and cardinality of the subset, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ [8,30].

4. The proposed hybrid BGSA with evolutionary operators

As stated in the literature review, GSA may still suffer from trapping in Local Optima (LO) when dealing with challenging problems. feature selections problems are very challenging due to the large number of variables and local solutions. The search space of FS problems changed for each dataset. Therefore, a simple but efficient strategy is highly demanded to mitigate the core drawbacks of the GSA: first, lack of a simple but efficient technique to switch from the extensive exploration step to focused exploitation phase. Second, searching with an unstable balance between exploration and exploitation. Third, searching with a slow convergence speed. The last drawback is dependent to the insufficient tendency of the algorithm to be devoted to exploitation phase in last iterations, while the first and second one can lead to stagnation behavior and immature convergence in often the middle of iterations, which can significantly decrease the quality of final solutions.

In order to cope with the above-mentioned shortcomings, an improved evolutionary variant of GSA is proposed for the first time in this section. The main constraint for us is to improve the performance of GSA in tackling FS problems substantially while preserving the core merit of the GSA, including cheap computational complexity. As such, we proposed three

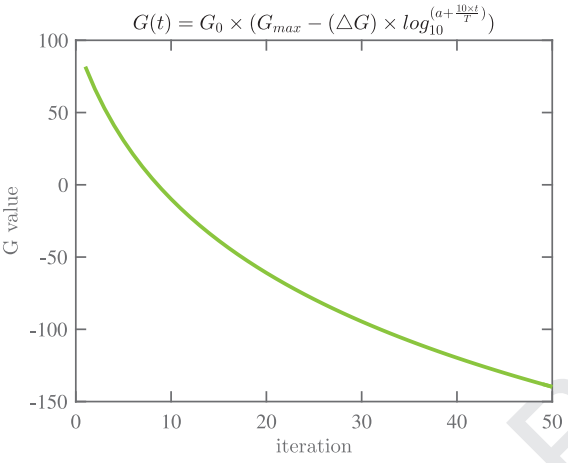

Fig. 4. Behavior of G'(t) over iterations. G_0 =10, $G_{min} = 0$, $G_{max} = 10$, $a = 1$, $T = 50$.

effective modifications to significantly boost the exploration and exploitation trends of the conventional GSA along with its convergence rate.

Firstly, a logarithmic decreasing function is adopted to gradually update the gravitational constant in the algorithm instead of the linear decreasing function. Secondly, a novel evolutionary mechanism, which mainly depends on a crossover scheme, is proposed to further improve the exploratory tendency of GSA. Here, the idea of the social thinking (gbest) in PSO was introduced, which has been integrated with the GSA algorithm [35]. Finally, a mutation operator is applied to assist the gbest solution in further improving the quality of results and avoiding the stagnation to LO. In the next subsections, the proposed operators are explained in detail:

4.1. Improving the convergence rate

In the basic GSA, the gravitational constant (G) is designed to control the right balance between the exploration and exploitation steps, where its value is dynamically changed during the optimization process as shown in Eq. (11). At the beginning of the process, it starts with a large value, which allows search agents to move with large steps. Small steps are allowed at the later stages when its value is linearly decreased. However, over the course of iterations, the search agents get heavier, and hence, the searching trend gets slower. This behavior can negatively affect the exploration and convergence speed of the GSA in dealing with some feature spaces. To overcome this drawback, we propose to utilize a logarithmic updating strategy, which is shown in Eq. (12), to replace the linearly decreasing one in the improved version of GSA. This time-varying logarithmic updating strategy can make a more stable balance between exploratory and exploitative steps and facilitate a smoother transition from broad exploration in initial steps to more focused exploitation at the later steps. This strategy has also been enhanced the convergence speed of the PSO in [14].

$$G(t) = G(t-1) \times \frac{t}{T}, \tag{11}$$

$$G'(t) = G_{t-1} \times \left(G_{max} - (\triangle G) \times log_{10}^{(a+\frac{10\times t}{T})}\right), \triangle G = G_{max} - G_{min} \tag{12}$$

where G_0 is the initial gravitational constant in range $[G_{min}, G_{max}]$, a is the acceleration rate suggested by Gao et al. [14] to set it to 1, t is the current iteration, and T is the total number of iterations. Fig. 4 shows the time-varying decreasing behavior of $G'(t)$ over iterations.

4.2. Improving the exploration by a crossover scheme

The original GSA depends on the mass of objects (solutions) to control the exploration tendency where the heavier solutions (superior solutions) are slower than the lighter ones (inferior solutions). Hence, the good solutions change their positions more slowly in order to reach better fitness values in the vicinity of the current areas. By contrast, the inferior solutions change their position with faster trends in order to discover new areas inside the feature space.

In order to further assist the GSA in the effective explore a feature space, we proposed a mechanism based on the crossover operator of genetic algorithms to generate new solutions and to enhance the exploratory powers of GSA. A simple example of a crossover operator for binary solutions is shown in Fig. 5. In the proposed mechanism, in each iteration, we sort the search agents based on their fitness values (mass of objects), while the first is the best. Then, we perform the crossover operator based on the last half of the population (worst solutions) and the global best agent in the solution until

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

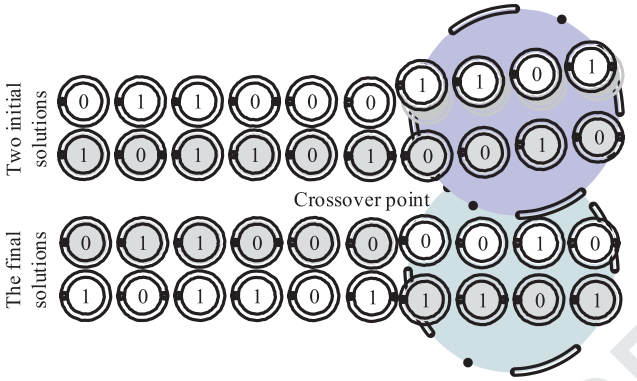

Fig. 5. Crossover operator.

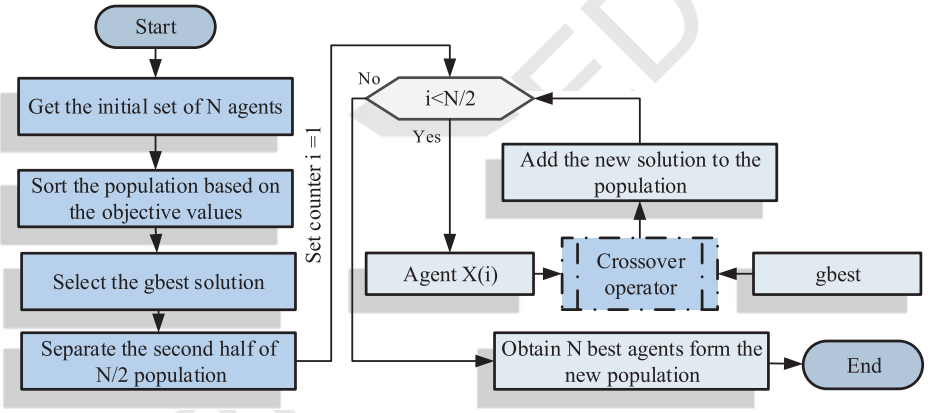

Fig. 6. General steps for exploration improvements.

the current iteration (gbest). Note that the definition of gbest solution here is similar to the α, leader of wolves in GWO or gbest in PSO. By performing the new mechanism, $\frac{N}{2}$ new extra solutions are generated and seeded into the pool of objects. Therefore, the size of the population becomes $N + \frac{N}{2}$. Finally, we take the best N solutions from the total $N + \frac{N}{2}$ solutions to be used in the next iteration. The steps of this process also shown in Fig. 6.

Note that this mechanism permits to avoid the worst misleading agents and enhance the quality of the population in a gradual manner. In addition, the crossover scheme can enhance the exploration merits of proposed GSA-based technique in the case of immature convergence and stagnation to LO.

4.3. Improving the exploitation by a mutation scheme

During the algorithm's life cycle, periodically, we check if the algorithm falls into LO. This is done by counting the number of times that the best solution obtained so far has not been improved during the optimization process. In that case, we applied the GA's mutation operator on the *gbest* to push it outside of the trap. Algorithm 3 summarizes the main steps of this process. Based on Algorithm 3, the generated solution from the mutation operation is reconsidered based on its fitness value. It might become the *gbest* when its fitness overcomes the *gbest*'s fitness or the fitness of local best agent *lbest*. In other words, the mutation will produce a new solution; the solution's fitness will determine the rule or the position of that solution in the main population. Note that the population is sorted in an ascending order based on the fitness values of agents. Hence, the best solution takes first place, i.e. $X(1)$, while the worst solution takes the last position i.e. $X(N)$. This mechanism further alleviates the entrapment shortcoming of GSA to avoid trapping into LO when the leader of the swarm (heaviest object) has an inertia to the local optima and there is no improvement in the quality of *gbest* agent. In addition, it also enriches the quality of solutions and exploitation tendency of the proposed method.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Algorithm 2 Pseudo-code of crossover operator.

```
function CROSSOVER(gbest,agents)
    Sort(agents,'fitness')
    for i = 1, i < N/2, i++ do
        agents(N + i) = CrossOver(agents[i], gbest)
    Sort(agents,'fitness')                          ▷ number of agents is N + N/2
    agents = agents[1 : N]                          ▷ number of agents is N
    return agents
```

Algorithm 3 Pseudo-code of mutation operator.

```
function MUTATION(gbest,agents)
    temp = GAMutation(gbest)
    if Fitness(Temp) < Fitness(gbest) then
        agents(N) = gbest
        gbest = temp
    else if Fitness(temp) < Fitness(lbest) then
        agents(N) = lbest
        agents(1) = temp
    else if Fitness(temp) < Fitness(worst) then
        agents(N) = temp
    return agents
```

5. Experimental result and discussion

This section presents the extensive results of the proposed approaches. The comparisons are conducted in three phases. In the first phase, the results of the HGSA were compared with the results of the binary BGSA approach. Since we used two classifiers for evaluation of the feature subsets, the results of each classifier were compared separately. Then, the results of HGSA on KNN classifier were compared to those obtained by HGSA with the DT classifier.

In the second phase, the results of HGSA were compared to the state-of-the-art approaches (i.e. GWO, PSO, and GA) where the two classifiers (i.e., KNN and DT) were used as well. The selected algorithms belong to different categories according to the nature of operators; GA [20] was selected as an evolutionary-based algorithm, PSO [23], GWO [33], and GSA were selected as swarm-based algorithms. Finally, the results of the HGSA were compared with the latest FS approaches from the literature. To determine whether there is a significant difference between the compared approaches or not, a statistical test based on the Wilcoxon rank-sum test was conducted. Wilcoxon test can be used to calculate the null hypotheses for two populations with the same continuous distribution. The p-value that is less than 0.05 could be considered as strong evidence against the null hypothesis.

5.1. Setup of experiments

In all experiments, a PC with Intel Core i7, and 8GB RAM was used. All the algorithms are performed on the same configurations and settings as illustrated in Table 4. Since the experiments focus on stochastic algorithms, we reported the average of the results for 30 runs on each dataset. Please note that the **bold** numbers in all subsequent tables denote the best values. Many researchers have recommended that $K = 5$ in KNN is the proper value to be used with the adopted datasets in this paper [8,30,32,43]. CART DT is used in this research. All algorithms in this work has same number of fitness evaluations therefore, the same number of iterations is applied for all.

To investigate the efficiency of the proposed approaches, eighteen different datasets from UCI repository [27] are utilized. The datasets are chosen with different properties (e.g., number of features, number of instances, and number of classes), and from various kinds of real-life problems. Table 1 reports a brief description (including the number of features and instances) of the utilized datasets.

5.2. Evaluation criteria

Three performance indicators are used for the comparative purposes; classification accuracy, number of selected features, and the fitness values.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Algorithm 4 Pseudo-code of HGSA algorithm.

Input: Number of agents and total number of iterations (L).
Output: The best agent and the its fitness value.
Initialize $G_0 = 10$ and $a = 1$ ▷ gravitational constant, acceleration ratio
Initialize the candidate agents $X_i (i = 1, 2, \ldots, N)$
Evaluate the fitness of all objects
while (end condition is not met) **do**
  `agents = Crossover(gbest,agents)` ▷ Call Algorithm 2
  if *gbest* is stuck **then**
    `agents = Mutation(gbest,agents)` ▷ Call Algorithm 3
  Evaluate the fitness of all agents
  Calculate *best*, *worst*, M_i, M_j
  Update the gravitational factor $G(t)$ using Eq. (12)
  Calculate forces F_{ij} by Eq. (6)
  Update acceleration by Eq. (5)
  Update velocities using Eq. (4)
  Update positions by Eq. (3)
Return the best search agent

Table 1
List of datasets.

Dataset	No. of Features	No.of instances
Breastcancer	9	699
BreastEW	30	569
Exactly	13	1000
Exactly2	13	1000
HeartEW	13	270
Lymphography	18	148
M-of-n	13	1000
PenglungEW	325	73
SonarEW	60	208
SpectEW	22	267
CongressEW	16	435
IonosphereEW	34	351
KrvskpEW	36	3196
Tic-tac-toe	9	958
Vote	16	300
WaveformEW	40	5000
WineEW	13	178
Zoo	16	101

- *Average classification accuracy*: this indicator measures how accurate is the classifier in predicting the right class using the selected subset of features. The average accuracy is calculated based on Eq. (13):

$$AvgAccuracy = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}(C_i == L_i) \tag{13}$$

where M is the number of runs for an algorithm to find the final subset of features, N is the number of dataset instances, C_i is the predicted class, and L_i is the actual class in the labeled data.

- *Average fitness*: This indicator is a composition from a classifier error rate (KNN or DT) and the features reduction rate. This indicator is used as an objective function in the optimization algorithm to identify the goodness of the selected feature subset. The average fitness value for each run is calculated as in Eq. (14):

$$AvgFitness = \frac{1}{M}\sum_{j=1}^{M} Fit_*^i \tag{14}$$

where M is the number of runs, Fit_*^i is the fitness of the best solution resulted from run i.

- *Average selection size*: This indicator represents the performance of an algorithm in terms of selection size when solving the FS problem. This indicator is calculated as in Eq. (15):

$$AvgSize = \frac{1}{m}\sum_{i=1}^{M}\frac{d_i^*}{D} \tag{15}$$

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Table 2
Analyzing the impact of population size and number of iterations on the fitness values.

Pop. Size/iteration	25	50	100	150	200
5	0.1007	0.0939	0.0871	0.0777	0.0646
7	0.0878	0.0908	0.0779	0.0771	0.0746
10	0.0847	0.0893	0.0657	0.0766	0.0694
20	0.0827	0.0708	0.0742	0.0682	**0.0603**
30	0.0733	0.0642	0.0738	0.0669	0.0892

Table 3
Analyzing the impact of K parameter and G_0 on the fitness values.

G_0/K	1	3	5	7	9
3	0.0551	0.1075	0.0553	0.1545	0.1022
7	0.0899	0.0323	0.1077	0.0905	0.1896
10	0.0379	0.0205	**0.0087**	0.1024	0.1775
25	0.0899	0.096	0.0961	0.0728	0.1662
30	0.0844	0.0903	0.0438	0.0845	0.1139

Table 4
Experimental setup .

Config. Name	Value
Number of runs	30
Number of iterations	200
Number of agents	20
G_0 (for GSA)	10
a (for GWO)	from 2 to 0
ω (for PSO)	from 2 to 0
GA selection	Roulette Wheel Selection
Mutation Probability (in GA)	50%
Crossover probability (in GA)	50%
K for KNN	5

where M is the number of runs, d_i^* is the number of selected features (turned on values) in the binary solution vector from the i-th run, and D is the total number of features in the original dataset.

5.3. Sensitivity analysis

This initial experiment is conducted to perform a sensitivity analysis and to study the influence of the main parameters of the proposed algorithm, which are the population size, number of iterations, G_0, and k. For this, PenglungEW dataset is selected to conduct the experiments because this dataset showed the highest sensitivity when we performed training of the classification models. The sensitivity analysis is performed in two stages. First, we study the effect of population size and maximum number of iterations, while the K and G_0 are fixed to 5 and 10, respectively. Table 2 shows the results of this stage. It can be seen from Table 2 that the best (i.e. smallest) fitness value is obtained when the population size is 20 and the number of iterations is 200.

In the second stage, the impact of k and G_0 are studied by varying their values while fixing the population size and the maximum number of iterations to the best values obtained in the previous stage, which are 10 and 200, respectively. Table 3 shows the results of this stage. It can be noticed from Table 3 that the best fitness value is reached when we have $G_0 = 10$ and $k = 5$. The best obtained values of these parameters will be used for rest of the experiments.

5.3.1. Comparing HGSA and BGSA with KNN and DT classifiers

Table 5 shows the average classification accuracy results for HGSA and BGSA. Inspecting the results in this table, it can be observed that HGSA performs better than BGSA in terms of classification accuracy when using KNN classifier, where it obtained the best results in 61% of the datasets, while BGSA was able to outperform HGSA in 38% of the datasets. Both optimizers obtained the same classification accuracy on one dataset (i.e. Exactly).

When using the DT classifier, HGSA is still better than BGSA. Again, HGSA obtained the best results in 61% of the datasets while BGSA performs better than HGSA in seven out of 18 datasets, and both algorithms shared the same accuracy (96.6%) for "WineEW" dataset. It is worth mentioning that HGSA was able to obtain the best results for both KNN and DT on the same eight datasets, while BGSA performed the same on four datasets only.

Number of selected features is another important aspect of wrapper FS approaches. Table 6 shows a comparison between BGSA and HGSA over the minimal number of selected features on all datasets. When analyzing the reported result, its can be clearly observed that HGSA significantly outperformed BGSA in minimizing the number of selected feature. As can be

Table 5
Comparison between BGSA and HGSA in terms of average classification accuracy for KNN and DT classifiers.

Dataset	KNN classifier		DT Classifier	
	BGSA	HGSA	BGSA	HGSA
Breastcancer	0.971	**0.974**	**0.968**	0.966
BreastEW	**0.973**	0.971	0.957	**0.964**
CongressEW	**0.968**	0.966	**0.981**	0.949
Exactly	**1.000**	**1.000**	0.807	**0.984**
Exactly2	0.763	**0.770**	0.761	**0.764**
HeartEW	**0.866**	0.856	**0.867**	0.815
IonosphereEW	0.908	**0.934**	0.947	**0.963**
KrvskpEW	0.974	**0.978**	0.991	**0.996**
Lymphography	0.886	**0.892**	**0.861**	0.835
M-of-n	0.986	**1.000**	**1.000**	0.992
penglungEW	0.939	**0.956**	0.726	**0.794**
SonarEW	0.890	**0.958**	0.848	**0.869**
SpectEW	0.893	**0.919**	0.893	**0.902**
Tic-tac-toe	**0.798**	0.788	0.849	**0.866**
Vote	0.970	**0.973**	0.947	**0.960**
WaveformEW	**0.818**	0.815	**0.783**	0.769
WineEW	**0.994**	0.989	**0.966**	**0.966**
Zoo	**0.998**	0.932	**0.958**	0.950

Table 6
Comparison between BGSA and HGSA in terms of average number of selected features for KNN and DT classifiers .

Dataset	KNN classifier		DT Classifier	
	BGSA	HGSA	BGSA	HGSA
Breastcancer	4.07	**4.00**	**2.60**	3.13
BreastEW	16.23	**13.80**	12.13	**7.10**
CongressEW	5.40	**3.93**	**2.70**	3.00
Exactly	**6.00**	**6.00**	6.70	**6.00**
Exactly2	**3.50**	4.70	**3.43**	3.70
HeartEW	**6.43**	6.70	3.07	**3.00**
IonosphereEW	10.90	**8.17**	13.57	**7.50**
KrvskpEW	16.97	**15.73**	22.03	**20.00**
Lymphography	7.40	**7.03**	4.83	**4.13**
M-of-n	7.00	**6.00**	**6.00**	7.00
penglungEW	150.07	**43.03**	154.07	**20.40**
SonarEW	26.00	**25.40**	27.93	**8.20**
SpectEW	11.60	**8.03**	8.50	**6.00**
Tic-tac-toe	8.87	**8.73**	**7.03**	8.00
Vote	9.00	**3.07**	**3.37**	4.00
WaveformEW	20.03	**19.50**	18.30	**16.43**
WineEW	6.23	**5.00**	5.07	**4.00**
Zoo	7.83	**5.63**	5.97	**4.93**

seen in Table 6, HGSA provided the best results in 89% of the datasets. However, BGSA outperformed HGSA on only three dataset including the m-of-n dataset where both approaches obtained the same result.

When using the DT classifier, HGSA also showed a superior performance over the BGSA approach. HBGSA was able to obtain the minimal reduct in 68% of the datasets, while BGSA obtained the best results in the remaining datasets. These results show the ability of the HGSA in finding the most important features by better exploring the search space than BGSA.

Table 7 shows the average fitness values that recorded for both HGSA and BGSA. Observing the fitness values of the KNN classifier in the table, it can be seen that the HGSA obtained better results than BGSA in ten out of eighteen datasets, while BGSA obtained the best results in seven datasets, and both algorithms performed the same result for the Exactly dataset. So, HGSA achieved the best results in 55% of the datasets. The same observation can be made when considering the results for the DT classifier, where HGSA outperform the BGSA in eleven out of eighteen datasets, while BGSA performs better in seven datasets. As such, HGSA is better than BGSA in 61.1% of the datasets.

The superior results show the merits of the proposed approach in both classification accuracy and reduction rate since both criteria are involved in the fitness function. This can be interpreted due to some strong properties of the GSA algorithm, and the enhancement we made in the HGSA approach. In GSA, the gravitational force, that is applied on each individual in the swarm, can be considered as knowledge-transferring tool. Thus, each individual can adjust its position based on some knowledge about the surrounding area and make prudent exploration and exploitation depending on its mass. Moreover,

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Table 7
Comparison between BGSA and HGSA in terms of average fitness values for KNN and DT classifiers.

Dataset	KNN classifier		DT Classifier	
	BGSA	HGSA	BGSA	HGSA
Breastcancer	0.034	**0.031**	**0.035**	0.038
BreastEW	**0.032**	0.033	0.046	**0.038**
CongressEW	**0.036**	0.036	**0.021**	0.052
Exactly	**0.005**	**0.005**	0.196	**0.021**
Exactly2	0.237	**0.232**	0.239	**0.237**
HeartEW	**0.138**	0.149	**0.135**	0.186
IonosphereEW	0.095	**0.068**	0.057	**0.039**
KrvskpEW	0.031	**0.027**	0.015	**0.010**
Lymphography	0.117	**0.111**	**0.141**	0.166
M-of-n	0.020	**0.005**	**0.005**	0.014
penglungEW	0.065	**0.045**	0.276	**0.204**
SonarEW	0.113	**0.046**	0.155	**0.131**
SpectEW	0.112	**0.084**	0.110	**0.100**
Tic-tac-toe	**0.211**	0.221	0.159	**0.142**
Vote	0.035	**0.028**	0.055	**0.042**
WaveformEW	**0.185**	0.189	**0.220**	0.233
WineEW	**0.011**	0.015	0.038	**0.037**
Zoo	**0.007**	0.071	**0.046**	0.052

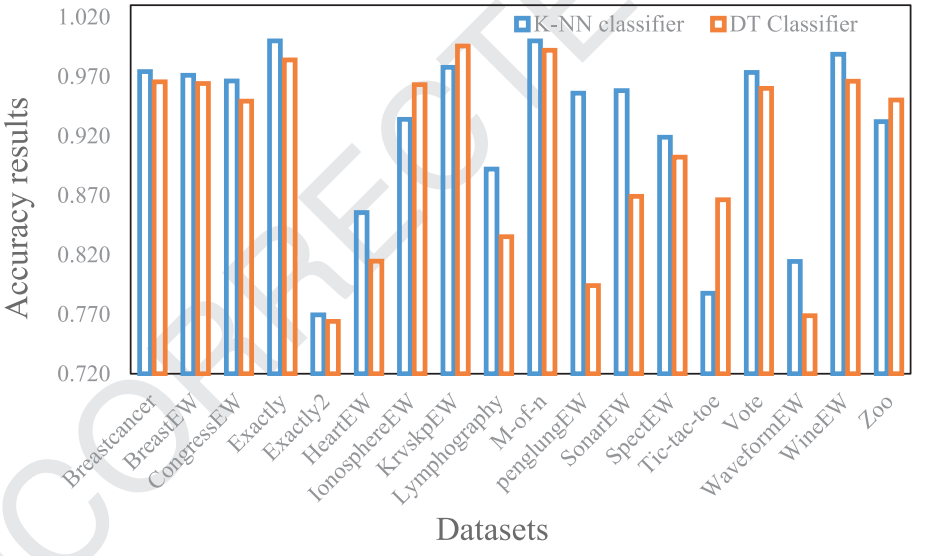

Fig. 8. Accuracy rates of HGSA-based wrapper with KNN and DT classifiers.

applying the evolutionary operator (i.e. crossover and mutation) in HGSA enhanced the GSA ability in escaping from LO and enhance the diversity of the algorithm.

Table 8 shows the best, average and worst results of BGSA and HGSA based on PenglungEW dataset which is the same dataset was selected in the previous section due to its high sensitivity. The table also shows the results for KNN and DT classifiers. It can be seen that HGSA outperforms BGSA in terms of best, average and worst results of fitness values, accuracy and number of selected features.

5.3.2. Comparing HGSA performance for KNN and DT

In the previous sections, we compared the performance of HGSA and BGSA over the considered evaluation criteria. As it was shown previously, HGSA outperformed BGSA on the majority of datasets based on all criteria. Here, we are interested to examine the performance of HGSA on both classifiers. Table 9 shows the average classification accuracy, fitness values, and number of selected features results of HGSA on both KNN and DT classifiers. The average classification accuracy of both classifiers is visualized using line chart in Fig. 8. Boxplots of accuracy rates are also demonstrated in Fig. 7.

Inspecting the results in Table 9, and Fig. 7, it can be noted that HGSA obtained higher accuracy when using the KNN in 78% of the datasets. In contrast, the HGSA recorded a superior performance when using DT in terms of number of features. This result can be justified due to the nature of each classifier and the way it adapt to classify the testing samples. KNN is

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Table 8
Best, average and worst results of BGSA and HGSA based on PenglungEW dataset.

Classifier	Metric	Avg. Fitness		Avg. Accuracy		Avg. no. of selected features	
		BGSA	HGSA	BGSA	HGSA	BGSA	HGSA
KNN	Best	0.033	0.001	0.971	1	135	34
	Average	0.065	0.045	0.939	0.956	150.067	43.033
	Worst	0.092	0.06	0.912	0.941	166	65
DT	Best	0.266	0.204	0.735	0.794	134	10
	Average	0.276	0.204	0.726	0.794	154.067	20.4
	Worst	0.296	0.205	0.706	0.794	171	32

Table 9
The performance of HGSA for both KNN and DT classifiers for all measures.

Dataset	Avg. Accuracy		Avg. no. of Features		Avg. Fitness	
	KNN	DT	KNN	DT	KNN	DT
Breastcancer	**0.974**	0.966	4.00	**3.13**	**0.031**	0.038
BreastEW	**0.971**	0.964	13.80	**7.10**	**0.033**	0.038
CongressEW	**0.966**	0.949	3.93	**3.00**	**0.036**	0.052
Exactly	**1.000**	0.984	**6.00**	**6.00**	**0.005**	0.021
Exactly2	**0.770**	0.764	4.70	**3.70**	**0.232**	0.237
HeartEW	**0.856**	0.815	6.70	**3.00**	**0.149**	0.186
IonosphereEW	0.934	**0.963**	8.17	**7.50**	0.068	**0.039**
KrvskpEW	0.978	**0.996**	**15.73**	20.00	0.027	**0.010**
Lymphography	**0.892**	0.835	7.03	**4.13**	**0.111**	0.166
M-of-n	**1.000**	0.992	**6.00**	7.00	**0.005**	0.014
penglungEW	**0.956**	0.794	43.03	**20.40**	**0.045**	0.204
SonarEW	**0.958**	0.869	25.40	**8.20**	**0.046**	0.131
SpectEW	**0.919**	0.902	8.03	**6.00**	**0.084**	0.100
Tic-tac-toe	0.788	**0.866**	8.73	**8.00**	0.221	**0.142**
Vote	**0.973**	0.960	**3.07**	4.00	**0.028**	0.042
WaveformEW	**0.815**	0.769	19.50	**16.43**	**0.189**	0.233
WineEW	**0.989**	0.966	5.00	**4.00**	**0.015**	0.037
Zoo	0.932	**0.950**	5.63	**4.93**	0.071	**0.052**

Table 10
Comparison between HGSA and the state-of-the-art FS approaches in terms of average classification accuracy.

Dataset	KNN Classifier					DT Classifier				
	GWO	PSO	GA	BGSA	HGSA	GWO	PSO	GA	BGSA	HGSA
Breastcancer	**0.980**	0.972	0.973	0.971	0.974	0.960	0.946	0.964	**0.968**	0.966
BreastEW	0.954	0.948	0.953	**0.973**	0.971	0.948	0.926	0.958	0.957	**0.964**
CongressEW	0.927	0.957	0.940	**0.968**	0.966	**0.982**	0.948	0.958	0.981	0.949
Exactly	0.731	0.994	0.748	**1.000**	**1.000**	0.752	0.913	0.759	0.807	**0.984**
Exactly2	0.758	0.743	0.761	0.763	**0.770**	0.759	0.751	0.760	0.761	**0.764**
HeartEW	0.853	0.795	0.832	**0.866**	0.856	0.809	0.808	0.834	**0.867**	0.815
IonosphereEW	0.893	0.883	0.876	0.908	**0.934**	0.940	0.927	0.924	0.947	**0.963**
KrvskpEW	0.955	0.965	0.930	0.974	**0.978**	0.984	0.987	0.953	0.991	**0.996**
Lymphography	0.831	0.759	0.815	0.886	**0.892**	0.816	0.779	0.825	**0.861**	0.835
M-of-n	0.910	0.993	0.877	0.986	**1.000**	**1.000**	0.970	0.883	**1.000**	0.992
penglungEW	0.868	0.661	0.861	0.939	**0.956**	0.681	0.449	0.643	0.726	**0.794**
SonarEW	0.795	0.763	0.853	0.890	**0.958**	0.797	0.709	0.802	0.848	**0.869**
SpectEW	0.833	0.810	0.864	0.893	**0.919**	0.842	0.818	0.852	0.893	**0.902**
Tic-tac-toe	0.790	0.783	0.759	**0.798**	0.788	0.815	0.832	0.850	0.849	**0.866**
Vote	0.940	0.933	0.931	0.970	**0.973**	0.932	0.958	0.925	0.947	**0.960**
WaveformEW	0.784	0.788	0.761	**0.818**	0.815	0.748	0.735	0.739	**0.783**	0.769
WineEW	0.980	0.937	0.967	**0.994**	0.989	0.952	0.951	0.964	**0.966**	**0.966**
Zoo	0.924	0.817	0.930	**0.998**	0.932	0.888	0.845	0.924	**0.958**	0.950

classified under the category of *lazy learner* since it does not build any model and depends on calculating the distances between objects. On the other hand, DT classifier comes under the category of *Eager Learners*, because it builds a classification model based on the training data, which is used to classify the unseen objects from the testing set. DT classifier depends on the information gain to build a classification tree. Therefore, the tree is built based on the information that can be gained from features. This property gives DT the ability to select the discriminatory features better than KNN.

Comparing the performance of both classifiers in terms of selected features and the classification accuracy in Table 9 and Figs. 7 and 8, we can conclude that selecting the most informative features (in DT case) would reduce the number of

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

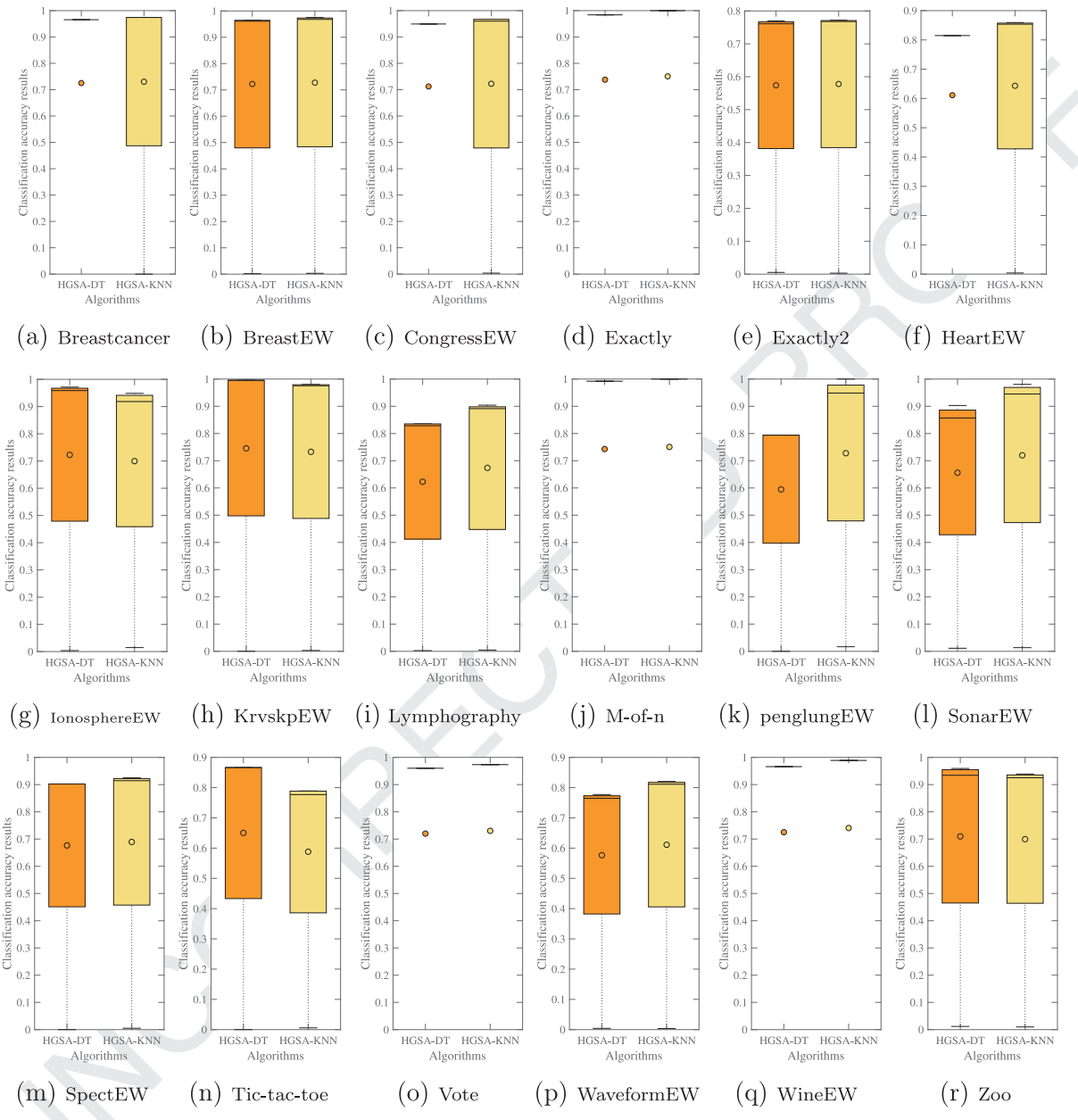

Fig. 7. Boxplots of accuracy rates of the proposed HGSA with DT and KNN classifiers for all datasets.

features, but it does not guarantee that those features reveal the highest classification accuracy. Since KNN does not consider the amount of information in each feature and consider the feature vector as a block, this improves its ability to find the best feature subset and reveal the highest accuracy.

The superiority of HGSA with KNN is observed again when considering the fitness values. Since there is more emphasize on the classification accuracy in the fitness function, these results are reasonable and expected. Its worth mentioning that the KNN-based approach is better than DT-based variant in terms of fitness values on 78% of datasets.

Figs. 9 and 10 compare the convergence behaviors of different methods. From Figs. 9 and 10, it can be observed that the convergence behaviors of HGSA is more accelerated than that of BGSA in general for both classifiers in harmony with the results in Table 7 when considering the KNN classifier. The same observation can be made by considering the DT classifier. It is also evident that the HGSA algorithm with KNN classifier converges faster than the variant with DT classifier in some datasets. Overall, it can be concluded that HGSA with KNN classifier shows the best convergence behavior on 10 datasets. The dissimilarities of the acceleration values in the experimented curves also reveal the substantial impact of the proposed enhancements, which is applied to the original GSA.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

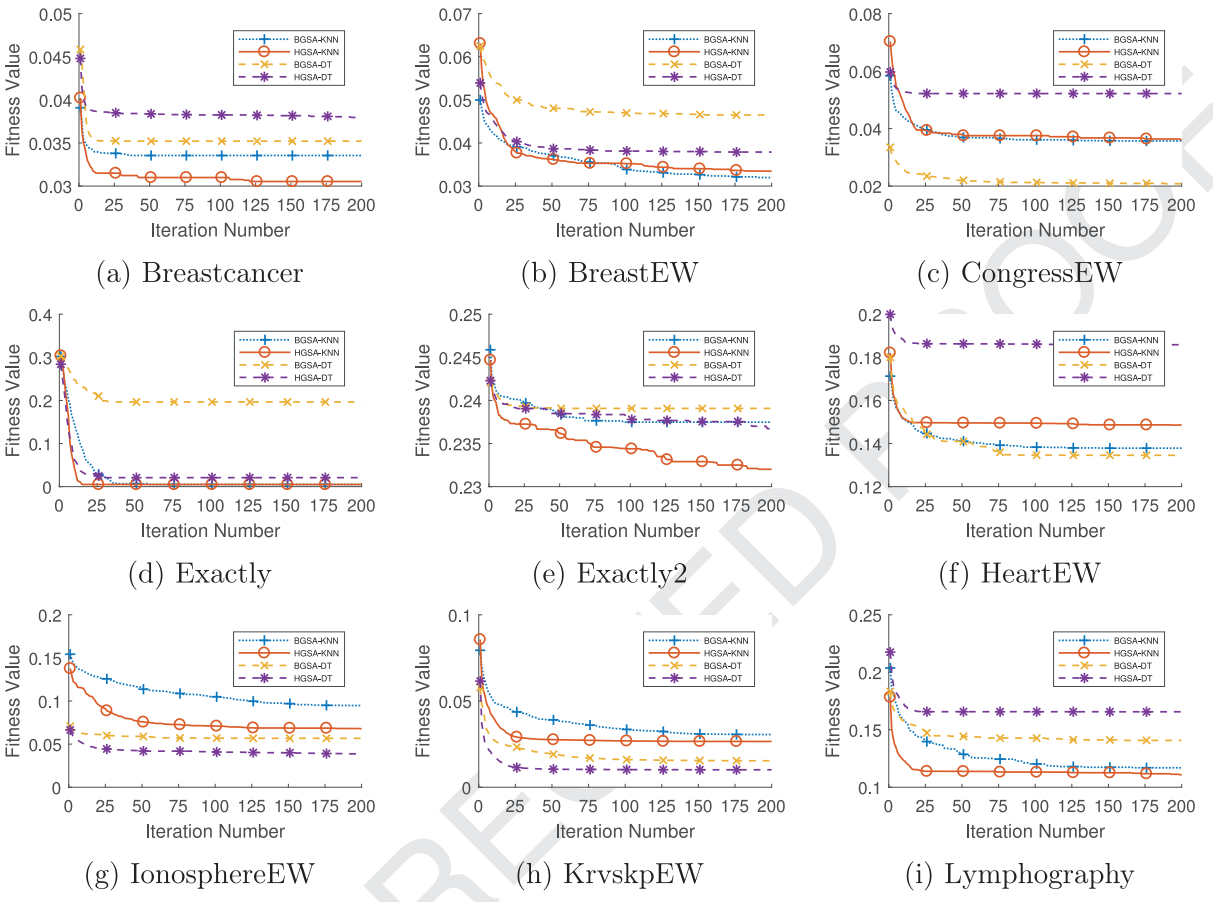

Fig. 9. Convergence curves for BGSA and HGSA on first 9 datasets with DT classifier.

5.3.3. Comparing HGSA with the-state-of-the-art approaches

After analyzing the results of BGSA and HGSA approaches with two KNN and DT classifiers, we found that the overall performance of HBGSA with both classifiers is much better than BGSA for all evaluation criteria on majority of datasets. The main purpose of this subsection is to compare the performance of HGSA with the state-of-the-art FS approaches; i.e. GA, PSO, and GWO. To make a fair comparison, all approaches were implemented for the comparison purposes with the same settings and conditions. Table 10 shows classification accuracy of different methods using both classifiers; KNN and DT. The results in Table 10 evidently show the exploratory and exploitative merits of the proposed HGSA using any classifier over other wrapper approaches. With the KNN classifier, HGSA outperformed other algorithms in sixteen out of eighteen (88.8%) datasets, while GWO obtained the best results only in two datasets. A similar observation can be made about HGSA using DT classifier, i.e. HGSA outperformed other algorithms in fifteen out of eighteen datasets (83.3%), while GWO with KNN performed better than other approaches in two datasets, and GA with DT only in one dataset. Table 11 shows the standard deviation values calculated for the accuracy results. It can be see that HGSA has very competitive small values of standard deviations compared to the other algorithms which indicates the stability of its performance. The p-values of Wilcoxon test is tabulated in Table 12. Results in Table 12 show that the superiority of HGSA is statistically significant because the most of p-values are less than 0.05 for both KNN and DT classifiers.

Table 13 compares the PSO, GA, GWO and HGSA based on average number of selected features. For both KNN and DT classifiers, HGSA obtained the best result on 83% and 88% of the datasets, respectively.

In terms of fitness values, Table 14 shows that HGSA with KNN outperformed other approaches on the majority of datasets. This becomes clear when we see HGSA outperformed other algorithms in sixteen out of eighteen (88.8%) datasets, while GWO in only two datasets. When we use DT, HGSA outperformed other algorithms in fifteen out of eighteen datasets (83.3%), while GWO performed better only in two datasets, and GA in single dataset.

The main reason for improved results and performance of the proposed GSA-based wrapper is that it has been integrated with a time-varying gravitational function in an adaptive manner, and can reveal improved crossover and mutation-based exploration and exploitation behaviors. Consequently, it can make a more stable balance between exploratory and exploitative inclinations when dealing with various feature spaces. As a result, the immature convergence and stagnation shortcom-

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

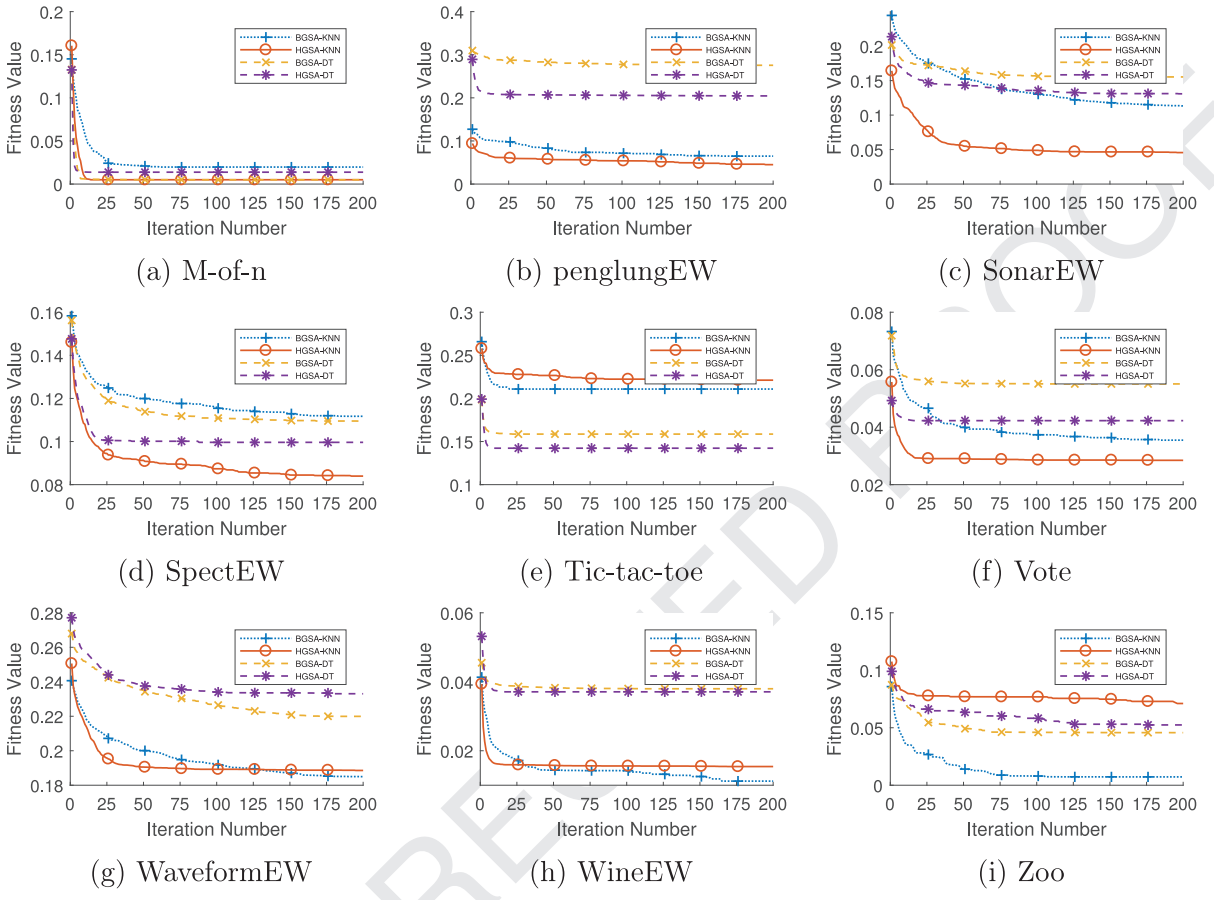

Fig. 10. Convergence curves for BGSA and HGSA on second 9 datasets with DT classifier.

ings of the basic GSA-based wrapper is alleviated, significantly. These evolutionary dynamic features has led to improved results in terms of accuracy rates, selected features, and fitness values.

5.3.4. HGSA Vs. state-of-the-art approaches

After analyzing the results of HGSA and comparing it with the state-of-the-art approaches, the results are also compared with other well-established wrapper approaches from the literature. Emary et al. [8] proposed GWO-based FS approaches. Their parameter settings are very similar to those we adopted. Assuming that the researchers used the best settings that assure the best results of their approaches, we obtained the results from their paper and made a comparison with the proposed HGSA. Since they used KNN classifier as an evaluator, we used to compare their work with the KNN based HGSA wrapper. Kashef and Nezamabadi-pour [22], is another research work where proposed another FS approach. The source code from the authors were used and the method also tested on the same datasets employed in this research.

Table 15 compares the proposed HGSA with KNN approach to GWO, PSO, and GA in terms of classification accuracy results. The results of these algorithms are taken from: GA1 in [22] PSO1 in [22] GWO1 in [8] GWO2 in [8] GA2 in [8] PSO2 in [8]. Please note that the algorithm names have been concatenated with numbers to distinguish them from each other. Fig. 11 also compares the accuracy rates of the proposed HGSA versus different methods. Observing results in Table 15 and Fig. 11, it can be clearly seen that HGSA can outperform other approaches in dealing with almost all datasets. The proposed wrapper obtained better results than GA1 method on all datasets. However, it is evident GWO2, GA2, and PSO2 are unable to show superior results in tackling any of these datasets.

There are some main reasons for the superiority of HGSA over other optimizers; first, the dynamic gravitational force, which is applied to each search agent, can be considered as a knowledge-transferring scheme. With this force, the particles can adjust their positions and scan more areas around themselves (vicinity of explored solutions) looking for better regions and solutions. In addition, the factor in Eq. (12) helps HGSA to start the search by more exploration and gradually and, then, more smoothly switch the broad exploration to more focused exploitative steps at the last iterations. This simple time-varying scheme brings several advantages to HGSA compared to other peers. For one, it helps HGSA to avoid premature convergence due to the hasty convergence to LO and limited exploration of the feature space. For another, it brings more

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Table 11
Comparison between HGSA and the state-of-the-art FS approaches in terms of standard deviation values .

Benchmark	KNN					DT				
	GWO	PSO	GA	BGSA	HGSA	GWO	PSO	GA	BGSA	HGSA
Breastcancer	0.004	0.006	0.001	0.001	0.000	0.003	0.011	0.002	0.003	0.000
BreastEW	0.005	0.008	0.005	0.003	0.003	0.005	0.012	0.004	0.002	0.002
CongressEW	0.008	0.014	0.005	0.001	0.004	0.000	0.015	0.002	0.002	0.000
Exactly	0.023	0.000	0.091	0.000	0.000	0.069	0.078	0.078	0.113	0.000
Exactly2	0.001	0.033	0.004	0.006	0.003	0.003	0.032	0.003	0.003	0.005
HeartEW	0.010	0.022	0.010	0.002	0.004	0.011	0.047	0.012	0.000	0.000
IonosphereEW	0.009	0.014	0.008	0.007	0.015	0.007	0.018	0.007	0.004	0.004
KrvskpEW	0.005	0.009	0.019	0.005	0.003	0.003	0.002	0.022	0.004	0.000
Lymphography	0.020	0.053	0.023	0.027	0.005	0.036	0.053	0.011	0.012	0.003
M-of-n	0.033	0.010	0.044	0.000	0.000	0.000	0.000	0.065	0.000	0.000
penglungEW	0.019	0.020	0.013	0.017	0.017	0.019	0.084	0.039	0.014	0.000
SonarEW	0.016	0.034	0.014	0.011	0.014	0.018	0.044	0.017	0.013	0.011
SpectEW	0.012	0.023	0.006	0.006	0.005	0.007	0.028	0.008	0.004	0.000
Tic-tac-toe	0.019	0.029	0.016	0.008	0.006	0.018	0.032	0.018	0.005	0.000
Vote	0.007	0.017	0.007	0.003	0.000	0.003	0.007	0.007	0.000	0.000
WaveformEW	0.003	0.009	0.009	0.003	0.004	0.005	0.010	0.008	0.005	0.004
WineEW	0.010	0.026	0.012	0.006	0.000	0.019	0.033	0.011	0.000	0.000
Zoo	0.017	0.044	0.014	0.006	0.010	0.015	0.041	0.011	0.007	0.012

Table 12
p-Values obtained from the rank-sum test for the results in Table 10 (Bold values are less than 0.05).

Dataset	KNN Classifier				DT Classifier			
	GWO	PSO	GA	HGSA	GWO	PSO	GA	HGSA
Breastcancer	**2.69E-09**	3.46E-01	**3.80E-06**	**2.71E-14**	**2.46E-11**	**7.21E-10**	**8.46E-04**	**7.29E-08**
BreastEW	**2.02E-11**	**2.12E-11**	**2.06E-11**	**5.25E-03**	**4.48E-12**	**3.96E-12**	**9.80E-09**	**9.95E-13**
Exactly	**3.68E-12**	6.15E-01	**2.71E-12**	1.51E-01	**1.69E-14**	6.40E-01	**1.18E-13**	**1.18E-13**
Exactly2	**1.16E-12**	**1.69E-14**	**4.47E-12**	**1.20E-12**	**3.31E-11**	**1.60E-13**	**1.20E-12**	**2.03E-05**
HeartEW	**8.60E-13**	3.68E-01	**1.61E-09**	**8.86E-06**	**2.28E-04**	**4.03E-02**	**6.34E-03**	**2.29E-02**
Lymphography	1.37E-01	**1.40E-11**	**1.24E-11**	**4.33E-12**	**4.19E-03**	6.41E-01	**5.17E-11**	**1.69E-14**
M-of-n	**1.07E-10**	**4.98E-11**	**2.33E-11**	**2.53E-08**	**1.42E-11**	**1.87E-11**	**1.35E-11**	**2.04E-11**
penglungEW	**2.32E-11**	**4.42E-10**	**2.33E-11**	**3.97E-03**	**3.03E-12**	**2.66E-12**	**3.14E-12**	**2.53E-12**
SonarEW	**3.26E-12**	**3.84E-12**	**3.21E-12**	5.40E-01	**1.01E-02**	**2.50E-05**	**8.63E-06**	**4.31E-12**
SpectEW	**1.20E-12**	**6.60E-04**	**1.20E-12**	**1.69E-14**	**1.69E-14**	**1.69E-14**	**1.09E-12**	**1.69E-14**
CongressEW	**8.28E-12**	**8.10E-12**	**4.19E-12**	**7.19E-04**	**4.16E-14**	**1.13E-12**	**5.22E-13**	**2.90E-13**
IonosphereEW	**2.36E-11**	**2.55E-11**	**2.25E-11**	**2.19E-11**	**1.81E-11**	**1.95E-11**	**3.80E-11**	**7.51E-08**
KrvskpEW	**1.66E-11**	**1.69E-11**	**1.03E-11**	**1.08E-11**	**5.90E-13**	**1.17E-12**	**8.74E-13**	**3.92E-12**
Tic-tac-toe	**7.09E-08**	4.95E-01	**1.87E-11**	**2.61E-12**	**1.04E-12**	**3.19E-11**	**1.52E-04**	**4.16E-14**
Vote	**8.30E-13**	**1.11E-12**	**8.26E-13**	**2.36E-05**	**1.18E-13**	8.15E-02	**3.77E-13**	**1.69E-14**
WaveformEW	**2.89E-11**	**2.95E-11**	**2.96E-11**	**1.59E-04**	**2.97E-11**	**2.98E-11**	**2.97E-11**	**1.58E-10**
WineEW	**4.65E-06**	**3.09E-11**	**3.95E-11**	**5.53E-05**	**7.44E-04**	1.56E-01	1.38E-01	**2.66E-12**
Zoo	**4.19E-02**	**2.99E-11**	4.51E-01	**9.62E-13**	**1.28E-11**	**1.07E-11**	**2.53E-09**	**1.03E-03**

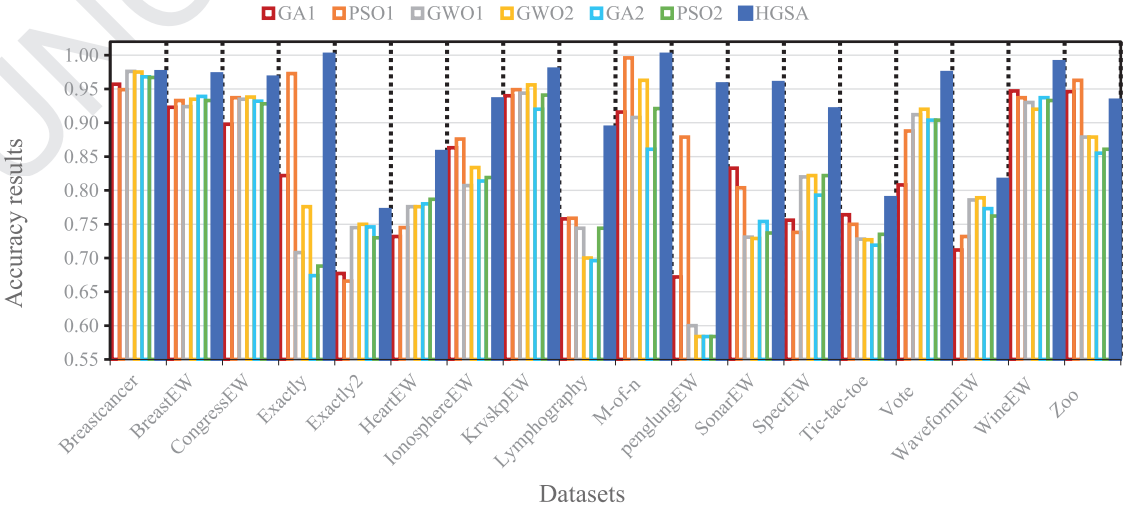

Fig. 11. Visual monitoring of average accuracy results.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

Table 13
Comparison between HGSA and the state-of-the-art FS approaches in terms of average number of feature .

Dataset	KNN Classifier					DT Classifier				
	GWO	PSO	GA	BGSA	HGSA	GWO	PSO	GA	BGSA	HGSA
Breastcancer	6.0	6.1	**3.9**	4.1	4.0	4.7	4.4	3.8	**2.6**	3.1
BreastEW	19.9	17.0	16.1	16.2	**13.8**	15.7	14.9	15.1	12.1	**7.1**
CongressEW	8.4	6.4	4.8	5.4	**3.9**	10.0	8.4	7.1	**2.7**	3.0
Exactly	10.0	**6.0**	7.4	**6.0**	**6.0**	7.4	6.2	7.3	6.7	**6.0**
Exactly2	**3.3**	4.6	4.1	3.5	4.7	3.6	5.7	4.1	**3.4**	3.7
HeartEW	11.7	7.9	6.9	**6.4**	6.7	7.9	6.2	6.4	3.1	**3.0**
IonosphereEW	18.2	14.5	15.2	10.9	**8.2**	20.9	17.4	15.6	13.6	**7.5**
KrvskpEW	26.8	19.3	19.5	17.0	**15.7**	30.3	26.4	20.2	22.0	**20.0**
Lymphography	8.9	9.2	7.6	7.4	**7.0**	8.4	9.1	8.3	4.8	**4.1**
M-of-n	10.0	6.3	7.9	7.0	**6.0**	9.2	9.1	7.7	**6.0**	7.0
penglungEW	159.8	161.7	151.3	150.1	**43.0**	162.2	161.9	155.3	154.1	**20.4**
SonarEW	31.9	29.9	29.5	26.0	**25.4**	35.0	28.6	28.9	27.9	**8.2**
SpectEW	11.6	10.6	11.2	11.6	**8.0**	14.0	11.4	10.9	8.5	**6.0**
Tic-tac-toe	8.5	7.1	**5.9**	8.9	8.7	7.3	7.4	7.2	**7.0**	8.0
Vote	11.2	6.6	7.4	9.0	**3.1**	8.0	8.0	6.2	**3.4**	4.0
WaveformEW	34.9	22.3	21.4	20.0	**19.5**	32.0	21.2	20.8	18.3	**16.4**
WineEW	7.8	7.6	6.9	6.2	**5.0**	6.7	7.5	6.2	5.1	**4.0**
Zoo	9.7	9.0	8.3	7.8	**5.6**	8.0	8.4	6.6	6.0	**4.9**

Table 14
Comparison between HGSA and the state-of-the-art FS approaches in terms of average fitness values.

Dataset	KNN Classifier					DT Classifier				
	GWO	PSO	GA	BGSA	HGSA	GWO	PSO	GA	BGSA	HGSA
Breastcancer	**0.027**	0.035	0.032	0.034	0.031	0.045	0.059	0.040	**0.035**	0.038
BreastEW	0.053	0.057	0.052	**0.032**	0.033	0.057	0.078	0.047	0.046	**0.038**
CongressEW	0.078	0.046	0.062	**0.036**	0.036	0.025	0.057	0.047	**0.021**	0.052
Exactly	0.275	0.011	0.255	**0.005**	**0.005**	0.251	0.090	0.244	0.196	**0.021**
Exactly2	0.242	0.258	0.240	0.237	**0.232**	0.241	0.250	0.241	0.239	**0.237**
HeartEW	0.155	0.209	0.172	**0.138**	0.149	0.196	0.195	0.170	**0.135**	0.186
IonosphereEW	0.112	0.120	0.128	0.095	**0.068**	0.066	0.078	0.080	0.057	**0.039**
KrvskpEW	0.053	0.040	0.075	0.031	**0.027**	0.024	0.021	0.052	0.015	**0.010**
Lymphography	0.173	0.244	0.188	0.117	**0.111**	0.187	0.224	0.178	**0.141**	0.166
M-of-n	0.097	0.011	0.128	0.020	**0.005**	0.008	0.037	0.122	**0.005**	0.014
penglungEW	0.136	0.340	0.142	0.065	**0.045**	0.320	0.551	0.358	0.276	**0.204**
SonarEW	0.209	0.240	0.150	0.113	**0.046**	0.207	0.293	0.201	0.155	**0.131**
SpectEW	0.171	0.190	0.140	0.112	**0.084**	0.163	0.185	0.151	0.110	**0.100**
Tic-tac-toe	0.218	0.223	0.246	**0.211**	0.221	0.192	0.174	0.157	0.159	**0.142**
Vote	0.067	0.071	0.073	0.035	**0.028**	0.072	0.047	0.078	0.055	**0.042**
WaveformEW	0.223	0.216	0.242	**0.185**	0.189	0.258	0.267	0.264	**0.220**	0.233
WineEW	0.026	0.068	0.038	**0.011**	0.015	0.054	0.054	0.041	0.038	**0.037**
Zoo	0.081	0.187	0.075	**0.007**	0.071	0.116	0.159	0.080	**0.046**	0.052

Table 15
Comparison of HGSA with KNN to other FS approaches with KNN from the literature in terms of average classification accuracy.

Dataset	GA1	PSO1	GWO1	GWO2	GA2	PSO2	HGSA
Breastcancer	0.957	0.949	**0.976**	0.975	0.968	0.967	0.974
BreastEW	0.923	0.933	0.924	0.935	0.939	0.933	**0.971**
CongressEW	0.898	0.937	0.935	0.938	0.932	0.928	**0.966**
Exactly	0.822	0.973	0.708	0.776	0.674	0.688	**1.000**
Exactly2	0.677	0.666	0.745	0.750	0.746	0.730	**0.770**
HeartEW	0.732	0.745	0.776	0.776	0.780	0.787	**0.856**
IonosphereEW	0.863	0.876	0.807	0.834	0.814	0.819	**0.934**
KrvskpEW	0.940	0.949	0.944	0.956	0.920	0.941	**0.978**
Lymphography	0.758	0.759	0.744	0.700	0.696	0.744	**0.892**
M-of-n	0.916	0.996	0.908	0.963	0.861	0.921	**1.000**
penglungEW	0.672	0.879	0.600	0.584	0.584	0.584	**0.956**
SonarEW	0.833	0.804	0.731	0.729	0.754	0.737	**0.958**
SpectEW	0.756	0.738	0.820	0.822	0.793	0.822	**0.919**
Tic-tac-toe	0.764	0.750	0.728	0.727	0.719	0.735	**0.788**
Vote	0.808	0.888	0.912	0.920	0.904	0.904	**0.973**
WaveformEW	0.712	0.732	0.786	0.789	0.773	0.762	**0.815**
WineEW	0.947	0.937	0.930	0.920	0.937	0.933	**0.989**
Zoo	0.946	**0.963**	0.879	0.879	0.855	0.861	0.932

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

stability to the searching performance of proposed algorithm when switching from exploration to exploitation. It can also emphasize on exploitation in last steps, which improves the quality of found feature sets.

6. Conclusion and future directions

In this work, a new hybrid GSA-based optimizer with evolutionary crossover and mutation schemes was proposed for the first time to tackle FS tasks. We substantiated both KNN and DT classifiers as the induction algorithms. Various problems were utilized to substantiate the efficacy of the proposed approaches. In order to study the quality of results, we compared the proposed HGSA with both KNN and DT versus other popular methods such GA, PSO, GWO, GSA and MFO algorithms. The extensive results and comparisons reveal the proposed HGSA can significantly outperform other wrapper methods and show merits in terms of exploration and exploitation, tradeoff between searching trends, and faster convergence rates compared to other peers on several FS tasks.

Finally, applying our novel algorithm (HGSA) to other domains, and using other FS models (e.g. filter approach using rough set theory), can be explored as a future extension of this work.

References

[1] R. Abbassi, A. Abbassi, A.A. Heidari, S. Mirjalili, An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models, Energy Convers. Manage. 179 (2019) 362–372.
[2] I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, Y. Zhang, S. Mirjalili, Asynchronous accelerating multi-leader salp chains for feature selection, Appl. Soft Comput. 71 (2018) 964–979.
[3] F. Caraffini, F. Neri, M. Epitropakis, Hyperspam: a study on hyper-heuristic coordination strategies in the continuous domain, Inf. Sci. 477 (2019) 186–202.
[4] F. Caraffini, F. Neri, G. Iacca, A. Mol, Parallel memetic structures, Inf. Sci. 227 (2013) 60–82.
[5] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Inf. Sci. 265 (2014) 1–22.
[6] Z. Chen, X. Yuan, H. Tian, B. Ji, Improved gravitational search algorithm for parameter identification of water turbine regulation system, Energy Convers. Manage. 78 (2014) 306–315.
[7] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE, 1995, pp. 39–43.
[8] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[9] H. Faris, A.M. Al-Zoubi, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah, H. Fujita, An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks, Inf. Fusion 48 (2019) 67–83, doi:10.1016/j.inffus.2018.08.002.
[10] H. Faris, I. Aljarah, M.A. Al-Betar, S. Mirjalili, Grey wolf optimizer: a review of recent variants and applications, Neural Comput. Appl. (2017) 1–23.
[11] H. Faris, M.M. Mafarja, A.A. Heidari, I. Aljarah, A.-Z. Ala'M, S. Mirjalili, H. Fujita, An efficient binary salp swarm algorithm with crossover scheme for feature selection problems, Knowl. Based Syst. 154 (2018) 43–67.
[12] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI Mag. 17 (3) (1996) 37.
[13] H.A. Firpi, E. Goodman, Swarmed feature selection, in: Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on, IEEE, 2004, pp. 112–118.
[14] Y.-l. Gao, X.-h. An, J.-m. Liu, A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation, in: Computational Intelligence and Security, 2008. CIS'08. International Conference on, 1, IEEE, 2008, pp. 61–65.
[15] J. Han, J. Pei, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.
[16] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, Y. Liu, Feature subset selection by gravitational search algorithm optimization, Inf. Sci. 281 (2014) 128–146.
[17] A.A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, S. Mirjalili, An enhanced associative learning-based exploratory whale optimizer for global optimization, Neural Comput. Appl. (2019), doi:10.1007/s00521-019-04015-0.
[18] A.A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, Soft. Comput. (2018) 1–18, doi:10.1007/s00500-018-3424-2.
[19] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. (2019).
[20] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[21] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, M.-H. Lim, Ockham's razor in memetic computing: three stage optimal memetic exploration, Inf. Sci. 188 (2012) 17–43.
[22] S. Kashef, H. Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, Neurocomputing 147 (2015) 271–279.
[23] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, 2011, pp. 760–766.
[24] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, 5, IEEE, 1997, pp. 4104–4108.
[25] C. Li, J. Zhou, Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm, Energy Convers. Manage. 52 (1) (2011) 374–381.
[26] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, X. Tian, An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis, Comput Math. Methods Med. 2017 (2017).
[27] M. Lichman, UCI machine learning repository, 2013.
[28] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, 454, Springer Science & Business Media, 2012.
[29] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, Binary dragonfly optimization for feature selection using time-varying transfer functions, Knowl. Based Syst. 161 (2018) 185–204, doi:10.1016/j.knosys.2018.08.003.
[30] M. Mafarja, I. Aljarah, A.A. Heidari, A.I. Hammouri, H. Faris, A.-Z. Ala'M, S. Mirjalili, Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, Knowl. Based Syst. (2017).
[31] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2017) 441–453.
[32] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.
[33] S. Mirjalili, I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, Grey Wolf Optimizer: Theory, Literature Review, and Application in Computational Fluid Dynamics Problems, Springer International Publishing, Cham, pp. 87–105. 10.1007/978-3-030-12127-3_6
[34] S. Mirjalili, A.H. Gandomi, Chaotic gravitational constants for the gravitational search algorithm, Appl. Soft Comput. 53 (2017) 407–419.
[35] S. Mirjalili, S.Z.M. Hashim, A new hybrid psogsa algorithm for function optimization, in: Computer and Information Application (ICCIA), 2010 International Conference on, IEEE, 2010, pp. 374–377.
[36] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.
[37] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1424–1437.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038

[38] Y. Pathak, K. Arya, S. Tiwari, Feature selection for image steganalysis using levy flight-based grey wolf optimization, Multimed. Tools Appl. (2018) 1–22.
[39] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. 297 (2015) 216–235.
[40] E. Rashedi, H. Nezamabadi-pour, Improving the precision of cbir systems by feature selection using binary gravitational search algorithm, in: Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on, IEEE, 2012, pp. 039–042.
[41] E. Rashedi, H. Nezamabadi-pour, Feature subset selection using improved binary gravitational search algorithm, J. Intell. Fuzzy Syst. 26 (3) (2014) 1211–1221.
[42] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Gsa: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[43] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Bgsa: binary gravitational search algorithm, Nat. Comput. 9 (3) (2010) 727–745.
[44] E. Rashedi, E. Rashedi, H. Nezamabadi-pour, A comprehensive survey on gravitational search algorithm, Swarm Evol. Comput. 41 (2018) 141–158, doi:10.1016/j.swevo.2018.02.018.
[45] S. Sarafrazi, H. Nezamabadi-Pour, S. Saryazdi, Disruption: a new operator in gravitational search algorithm, Scientia Iranica 18 (3) (2011) 539–548.
[46] B. Shaw, V. Mukherjee, S. Ghoshal, A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, Int. J. Electr. Power Energy Syst. 35 (1) (2012) 21–33.
[47] E.-G. Talbi, Metaheuristics: from Design to Implementation, 74, John Wiley & Sons, 2009.
[48] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[49] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intell. Syst. Appl. 13 (2) (1998) 44–49.
[50] M. Yin, Y. Hu, F. Yang, X. Li, W. Gu, A novel hybrid k-harmonic means and gravitational search algorithm approach for clustering, Expert Syst. Appl. 38 (8) (2011) 9319–9324.

Please cite this article as: M. Taradeh, M. Mafarja and A.A. Heidari et al., An evolutionary gravitational search-based feature selection, Information Sciences, https://doi.org/10.1016/j.ins.2019.05.038