



Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach

Ibrahim Aljarah¹ · Majdi Mafarja² · Ali Asghar Heidari^{3,4} · Hossam Faris¹ · Seyedali Mirjalili⁵

Received: 7 May 2018 / Revised: 15 March 2019 / Accepted: 24 March 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Grey wolf optimizer (GWO) is known as one of the recent popular metaheuristic algorithms inspired from the social collaboration and team hunting activities of grey wolves in nature. This algorithm benefits from stochastic operators, but it is still prone to stagnation in local optima and premature convergence when solving problems with a large number of variables (e.g., clustering problems). To alleviate this shortcoming, the GWO algorithm is hybridized with the well-known tabu search (TS). To investigate the performance of the proposed hybrid GWO and TS (GWOTS), it is compared with well-regarded metaheuristics on various clustering datasets. The comprehensive experiments and analysis verify that the proposed GWOTS shows an improved performance compared to GWO and can be utilized for clustering applications.

Keywords Optimization · Grey wolf optimizer · GWO · Tabu search · Data clustering

✉ Ibrahim Aljarah
i.aljarah@ju.edu.jo
<http://evo-ml.com>

Majdi Mafarja
mmafarja@birzeit.edu

Ali Asghar Heidari
as_heidari@ut.ac.ir; aliasgha@comp.nus.edu.sg; t0917038@u.nus.edu

Hossam Faris
hossam.faris@ju.edu.jo
<http://evo-ml.com>

Seyedali Mirjalili
seyedali.mirjalili@griffithuni.edu.au
<http://www.alimirjalili.com>

- ¹ King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
- ² Department of Computer Science, Birzeit University, West Bank, PO Box 14, Birzeit, Palestine
- ³ School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran
- ⁴ Department of Computer Science, School of Computing, National University of Singapore, Singapore, Singapore
- ⁵ School of Information and Communication Technology, Griffith University, Nathan, Brisbane, QLD 4111, Australia

1 Introduction

Nature-inspired algorithms mostly fall under a wider umbrella of optimizers called metaheuristics. In the last two decades, there has been an increased interest in the applications of metaheuristics in different machine learning tasks such as optimizing the learning process of classification and clustering algorithms [10,48]. In fact, metaheuristics have shown high capability in finding satisfactory solutions for challenging problems in a very reasonable time [1,19,49]. There are mainly two types of metaheuristics: population-based and trajectory-based algorithms [9,20,29,31,32,50,53]. The former type relies on generating a set of possible solutions for a given problem and improving those solutions using different operators to find an acceptable solution.

The population-based algorithms try to make a balance between exploring different regions of the search space (exploration) and local search (exploitation). Achieving a good trade-off between these two conflicting processes is necessary to obtain better optimization results for challenging problems [28,30]. In contrast to population-based algorithms, trajectory-based algorithms rely on updating just a single solution in their optimization cycle. Due to the nature of trajectory-based algorithms, local search around possible optimal solutions is promoted.

Popular examples of population-based metaheuristics are genetic algorithms (GA) [17], particle swarm optimization (PSO) [41], differential evolution (DE) [70], while some popular examples of trajectory algorithms include tabu search (TS) [22] and simulated annealing (SA) [43].

An important application of nature-inspired algorithms is in the field of clustering. Clustering can be considered as one of the most common unsupervised machine learning tasks. It is described as partitioning a set of unlabeled data instances into several subgroups (i.e., clusters), where points in the same subgroup are expected to have some similar pattern or behavior and be different from other points in other subgroups. Clustering has a wide spectrum of real-world applications like in spatial analysis, location-based services, transportation, text analysis, marketing, engineering, and image segmentation [34,77]. Although clustering is important in many applications, the problem is difficult to solve as an NP-hard problem.

There is a large number of proposed algorithms for data clustering in the literature. Most of these algorithms fall mainly within one of two categories: hierarchical clustering algorithms and partitional clustering algorithms. Hierarchical methods can cluster the data by merging smaller clusters into larger ones (agglomerative approach) or by splitting large clusters into smaller ones (divisive approach). On the other hand, partitional methods cluster data directly into disjoint clusters, while optimizing evaluation criteria over a predefined number of iterations [6,8,12]. Example of such criteria is to minimize the within-cluster sum of squares in the popular k -means partitional clustering algorithm. Main advantages of the partitional clustering algorithms are the simplicity and low computational cost. However, the main disadvantages of most partitional clustering algorithms are a high dependency on the initial solution and the probability of being trapped in local optimum solutions.

The problems of partitional clustering have been tackled in the literature in numerous studies. Researchers have proposed different approaches to overcome these problems. One of these popular approaches is the nature inspired based on partitional clustering [4,55]. In this work, one of the main contributions is to develop an effective clustering algorithm using the nature-inspired algorithm.

One of the recent metaheuristic algorithms that have gained remarkable attention in the optimization community is the grey wolf optimizer (GWO) algorithm. GWO was first pro-

posed in 2014 by Mirjalili et al. in [54]. The GWO mimics the ideal hunting behaviors of wolf packs. Since the release of its code in MATLAB and Python [18,54], the GWO has demonstrated very promising results when applied in various real-world applications [44,52,69]. Some of its advantages include its simplicity, few parameters to tune, unique population hierarchy, and the smooth transition from the exploration phase to the exploitation.

This paper presents a new hybrid approach based on GWO and the trajectory-based search algorithm TS for clustering. In this work, we have made the following key contributions:

- The proposed hybrid approach makes the use of TS to enhance the efficacy and the balance between the exploratory and exploitative behaviors of the GWO algorithm. To achieve this goal, the TS algorithm is deployed as an operator in GWO to search the neighborhood of the leaders.
- The hybrid GWO with trajectory TS algorithm is proposed for the first time to solve the partitional clustering tasks.
- The proposed algorithm has been tested on real datasets with different settings and characteristics to demonstrate its effectiveness and quality of solutions.

The method proposed in this paper has some potential advantages compared to previous population-based clustering methods: first, it has a higher local optima avoidance potential in the case of stagnation problems. The proposed GWO-based method can perform a more stable balance between exploratory and exploitative trends compared to the conventional methods and other comparable evolutionary methods with a leadership structure. Second, this method is locality-informed approach; that is, in the case of finding a high-quality solution, it can deeply scan the neighborhood regions with a structural step-by-step process based on well-known TS algorithm. Third, in view of the diversification and intensification mechanisms, this method utilizes a TS-based strategy to exploit the neighborhood of the leader wolves, which enrich the quality of cluster vectors found in previous stages. As this is the core engine of the proposed clustering method, all exploitative advantages are valid for partitional clustering scenarios.

This paper is structured as follows: Sect. 3 briefly describes the preliminaries of this work including the TS and GWO algorithms. In Sect. 5, the details of the proposed GWOTS for clustering problems are provided. The experiments and results are presented in Sect. 6. Finally, the findings of this work are reported and some possible research directions are suggested in Sect. 7.

2 Background

Evolutionary clustering-based algorithms are considered as one of the most successful and applied types of clustering techniques in the last two decades. One of the earliest examples of this type is GA-based evolutionary clustering. Maulik and Bandyopadhyay [51] developed a new clustering technique based on genetic algorithm (CGA) in 2000. The exploratory and exploitative traits of GA were employed to discover the best centroids. The attained results verified that CGA can outperform the traditional K -means technique. Agusti et al. [2] also proposed another version of GA called grouping GA (GGA). GGA is developed based on applying the grouping encoding and evolutionary crossover and mutation operators on the clustering process. The efficiency of the GGA shows that it can attain competitive accuracies. Other variants are also available in [13,14,61,64]. A particle swarm optimization (PSO)-based clustering approach was developed in [74]. The searching capabilities of PSO were utilized to obtain the best clusters. To utilize the merits of PSO like fast convergence speed in early

steps, [3] proposed a hybrid variant of PSO and K -means techniques. Parallel implementation of PSO-based clustering method was introduced in [5–8,26] proposed an evolutionary clustering strategy according to the searching merits the firefly algorithm (FA). In this paper, they proposed a hybrid FA- K -means technique. [47] proposed another evolutionary-based approach to tackle 2D spatial datasets. The assessment and comparisons revealed promising results for this technique.

In [66], ant colony optimization (ACO) was utilized to realize the best clusters. The results revealed the superior capabilities of this method compared to previous works. Aljarah and Ludwig [7] utilized glowworm swarm optimization (GSO) to realize the clustering tasks. In this work, the CGSO was modified and three objectives were employed to find several best centroids inside the solution space. The experiments revealed that CGSO-based technique can outperform four well-established algorithms on used datasets. In [58], a new swarm-based clustering technique has been developed based on binary artificial bee colony (BABC). Their motivation was to propose a new solution generation approach for dynamic clustering tasks in an effective manner based on a series of similarity metrics. Shukri et al. [68] also proposed evolutionary static and dynamic clustering techniques based on multi-verse optimizer (MVO). The MVO-based methods validated using 12 problems and the performance of this approach compared to several classic and population-based clustering strategies. The obtained results revealed that static and dynamic MVO-based solvers can beat other competitors on used datasets.

The GWO has also been utilized to deal with clustering scenarios. Fahad et al. [16] proposed a GWO-based clustering approach to deal with vehicular ad hoc networks (VANETs). The results show the improved results compared to some advanced variants of PSO such as comprehensive learning PSO (CLPSO). Katarya and Verma [39] proposed a movie-based collaborative recommender technique that employs the GWO and fuzzy c-mean (FCM) clustering approach to predict the score of an item for a customer with regard to his historical activity and similarity measures. Kumar et al. [45] proposed a GWO-based technique for clustering problems. The results were compared with different metaheuristic methods such as GA and PSO, and the outcomes of the simple GWO-based method were satisfactory. Kapoor et al. [37] also developed a GWO-based clustering method to deal with satellite image segmentation problems. For clustering analysis, Zhang and Zhou [78] hybridized the basic GWO with Powell local search, which is called PGWO, and the clusters revealed the superior results of PGWO. To deal with large-scale scenarios, Tripathi et al. [73] proposed a map-reduce-based modified GWO (MR-EGWO). Simulation results verified that the developed MR-EGWO can show a promising performance. All aforementioned studies on GWO-based approaches agree on the high potential and satisfactory performance of the conventional GWO in tackling clustering datasets.

There are also some other nature-inspired algorithms utilized for clustering objectives that can be read in [5,25,27,56,65].

3 Preliminaries

3.1 The GWO algorithm

The GWO is a recent swarm intelligence (SI) technique that was originally proposed by Mirjalili et al. [54]. The GWO has been successfully applied to many optimization problems [33,35,42]. The main inspiration of this algorithm came from the social behavior of the grey

wolves and their dominant hierarchy. In nature, wolves can often be seen in the packs with 5 to 12 individuals on average. Usually, two wolves (a dominant wolf and his mate) lead the folk, which is called alpha (α), and other pack's adult wolves follow them in the second level, which is called beta (β), while delta (δ) wolves come at the third level. Other wolves come in the lowest level and are called omega (ω). Alpha wolves are often responsible for guiding the hunting attacks, decision making for the main activities of the pack such as hunting, maintaining discipline, sleeping places, and waking time. Beta classes play the role of the advisors for alpha wolves and send the feedback from the other wolves to them. Delta members are responsible for guarding and protecting the pack from any danger, and delta group contains the scouts, sentinels, elders, hunters, and caretakers. Group hunting is another interesting behavior of the grey wolves. Wolves first track, chase, and approach a prey, and then, they pursue encircle and harass it until it stops moving. Finally, the wolves attack the stationary prey.

The social intelligence of grey wolves and their hunting mechanisms (tracking, encircling, and attacking the prey) are mathematically modeled to design the GWO algorithm. The social behavior is mathematically modeled to solve various problems by assigning the fittest solution of the population as α and the next two best solutions as β and δ , respectively. The remaining solutions in the population are called ω . The hunting process is modeled by simulating two processes: encircling a prey and then hunting it.

3.2 Encircling a prey

Encircling a prey by the hunters is the first step in the hunting process. The distance between each wolf and the prey can be modeled as in Eq. (1)

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \tag{1}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{2}$$

where $\vec{X}_p(t)$ indicates the position vector of the prey, $\vec{X}(t)$ indicates the position vector of a wolf, and t indicates the iteration. \vec{C} is calculated as in Eq. (1), and \vec{r}_2 is a random vector (of the same dimensions as $\vec{X}_p(t)$ and $\vec{X}(t)$) in the interval $[0, 1]$. The \cdot between \vec{C} and $\vec{X}_p(t)$ means corresponding componentwise multiplication.

3.3 Hunting a prey

When hunting a prey, wolves should get closer to the prey. Based on the distance of each wolf to the prey [\vec{D} in Eq. (1)], the next move of a wolf can be modeled as in Eq. (3).

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{3}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{4}$$

where \vec{a} should be linearly decreased from 2 to 0 over the optimization, and r_2 is a random vector inside $[0, 1]$.

Regarding the social hierarchy of wolves, the best three hunters suppose to have more knowledge for determining the location of the prey more than the wolves in the ω group.

Therefore, all wolves should update their positions according to the positions of α , β , and δ . The distance between any wolf and the leaders of the pack can be calculated using Eq. (5).

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \tag{5}$$

The obtained distances from Eq. (5) are utilized to reposition a wolf using Eqs. (6) and (7).

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \end{aligned} \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{7}$$

By repeating the encircling and hunting operators, the best location (prey) can be located.

As shown in Eq. (4), decreasing the values of \vec{a} from 2 to 0 means that the values of \vec{A} are random numbers in the interval $[-2a, 2a]$. When the values of \vec{A} are inside $[-1, 1]$, then a search agent can update its position to any location between his current position and the prey's position. The motions of wolves when $|A| < 1$ or $|A| > 1$ are shown in Fig. 1. The 3D view of motion is also depicted in Fig. 2. When $|A| > 1$, the wolves will check the unexplored space for finding any better preys. Hence, $|A| > 1$ has an effective role during the exploration phase of the GWO. It encourages all agents to still search the entire space in the first half of iterations for exploring better areas nearby the optimum. When $|A| < 1$, the wolves try to focus on the observed prey; hence, it assists the GWO in exploiting the neighborhood of solutions. It can decrease the size of jumps around the prey. The C vector also assists GWO during all iterations in escaping from the local optima (LO) in the situation

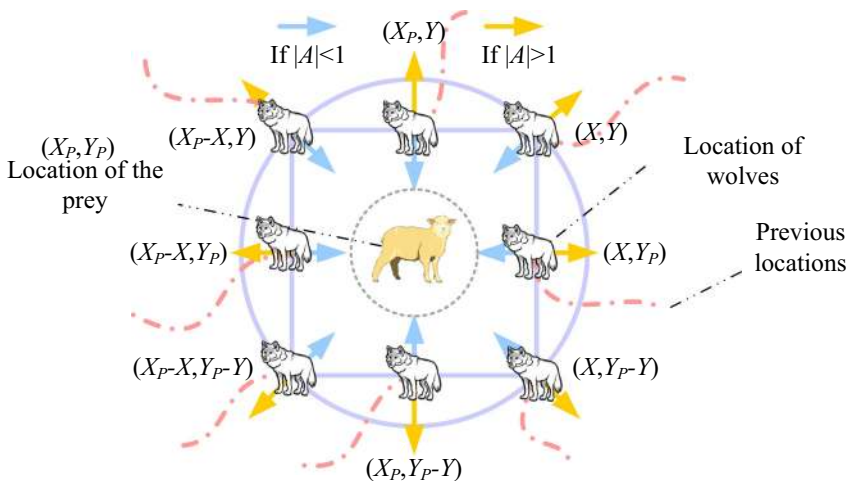


Fig. 1 Effect of A on the direction of motions

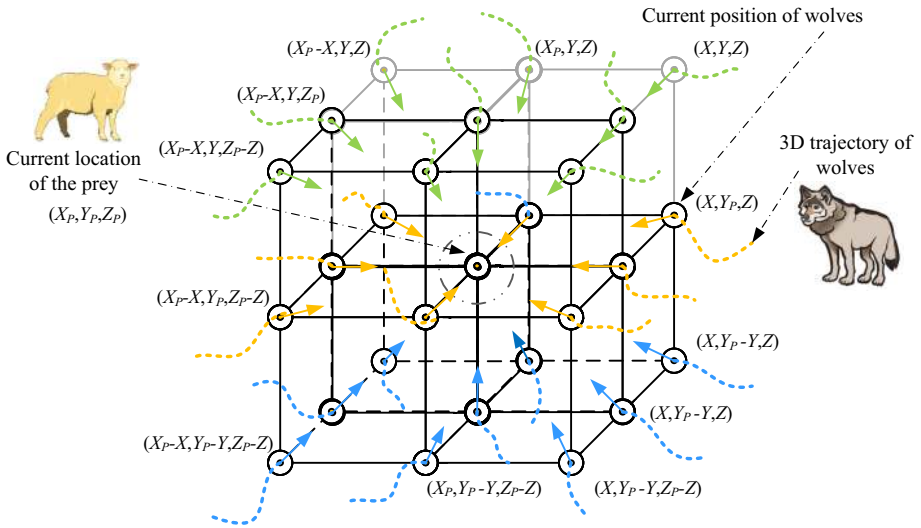


Fig. 2 3D motion of agents around the victim

of immature convergence. It enriches the random behaviors of GWO in dealing with more multifaceted landscapes. GWO intrinsically pulls all wolves toward the prey, which promotes exploitation. The random vectors can help GWO to make an effective transition from global to local searching phase. The pseudocode of the GWO is presented in Algorithm 1.

Algorithm 1 Pseudocode of the GWO

```

set the maximum number of iterations  $L$ 
Initialize the population  $X_i (i = 1, 2, \dots, n)$ 
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of wolves
 $X_\alpha$  = the best search agent
 $X_\beta$  = the second best search agent
 $X_\delta$  = the third best search agent
while ( $t < L$ ) do
  for each search agent do
    Update the position of the current search agent by Eq. (5)
  end for
  Update  $a$ ,  $A$ , and  $C$ 
  Calculate the fitness of all search agents
  Update  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ 
   $t = t + 1$ 
end while
return  $X_\alpha$ 
    
```

3.4 Tabu search (TS)

Tabu search (TS) is a metaheuristic algorithm that was originally proposed by Fred Glover [21]. The main feature of TS is that it uses an adaptive memory (called tabu list) to avoid returning to recently visited solutions to prevent the search from stacking at the local optima [72]. A key element of the tabu list is to create a balance between search exploration (diversi-

fication) and exploitation (intensification) [23]. Exploitation strategies are based on searching the attractive regions in the search space (the neighborhood of good solutions). Exploration strategies, on the other hand, aim to explore new regions in the search space that were not examined in the search process previously.

TS behaves like a hill-climbing algorithm (local search), but it accepts non-improving solutions to escape from local optima [22]. In TS, usually the whole neighborhood is explored and the best moves that improve the best solution so far are selected. In each iteration of the simple TS algorithm as shown in Algorithm 2, many trial solutions are generated in the neighborhood of the best solution. This generation process is designed to avoid the recently visited solutions. The best trial solution will be adopted as the next solution. The TS can be terminated if the number of iterations without any improvement exceeds a predetermined maximum iteration number. There are several works that hybridized the TS as a local search strategy with other optimizers such as PSO [67], SA [57], and ant colony optimization (ACO) [38].

Algorithm 2 Pseudocode of Tabu Search (TS)

```

 $S_{best} = S_0$  | *Construct Initial Solution* |
 $bestCandidate = S_{best}$ 
TabuList  $\leftarrow$  [ ] | *Initialize TabuList* |
while (NOT StoppingCondition ()) do
    Generate candidate solutions in the neighborhood of  $S_{best}$ 
    set  $SCandidate$  as the first candidate in the  $S_{best}$  Neighborhood
    for ( $SCandidate$  in  $S_{best}$  Neighborhood) do
        if ( $SCandidate$  NOT in TabuList AND  $fitness(SCandidate) > fitness(bestCandidate)$ ) then
             $bestCandidate \leftarrow SCandidate$ 
        end if
    end for
    if ( $fitness(bestCandidate) > fitness(S_{best})$ ) then
         $S_{best} \leftarrow bestCandidate$ 
    end if
    Update TabuList (push the  $bestCandidate$ )
    if ( $tabuListSsize > maxTabuListSize$ ) then
        Remove the first element from the TabuList
    end if
end while
return  $S_{best}$ 

```

4 The proposed hybrid GWOTS

The efficacy of the GWO is highly affected by its exploratory factors and exploitative hunting motions. It has a sufficient exploration potential and can find high-quality solutions compared to several well-established optimizers. When the algorithm explores a series of fruitful locations that have more chance to be the global best, the exploitation mechanism of GWO helps it to further focus on those solutions. However, if the wolves move toward the leader wolves during exploitation, it cannot scan the neighborhood of leader before updating all wolves in their direction. This situation happens when the algorithm needs a deep exploitation with occasional ‘jumps,’ but it cannot further increase the quality of solutions. After that, the algorithm cannot focus on exploitation of the neighborhood of best wolves found so far, which decreases the convergence rate and optimality of final results.

In GWO, the movement of ω wolves (the whole population except the leaders) highly relies on the situations of the leaders α , β , and δ . Since alpha, beta, and delta are selected to be the best three solutions obtained so far over the course of iterations, the leaders tend to converge to the same solution that might result in trapping in a local solution. To alleviate this drawback, a TS-based effective operator is considered here to be integrated into the GWO.

The key idea behind the hybrid GWOTS is to utilize the concept of adaptive memory in TS for discovering the neighborhood of each leader before updating the positions of the rest of the pack. Therefore, the TS algorithm is applied only to the locations of the leader wolves. The GWOTS is equipped with the adaptive memory and responsive exploration advantages of the TS. The role of the new TS-based operator is as an exploratory local search (ELS) that first receives a position of a leader. Then, it starts searching some attractive regions in the neighborhood space of the received position while considering a tabu list of recently visited positions to avoid returning of GWOTS to them, which consequently prevents the search from stacking at local solutions and goes toward better solutions. The movement mechanism of GWOTS is also demonstrated in Fig. 3.

Due to the use of TS inside the main GWOs loop, its computational complexity directly impacts that of GWO. The computational complexity of GWO is of $O(t_1 dn^2)$, where t_1 indicates the number of iterations, d is the number of variables, and n shows the number of solutions. In addition, the computational complexity of the tabu search (TS) is of $O(t_2 s)$, where t_2 is the number of iterations and s is the neighborhood size. TS is run in each iteration for three of the GWO solutions, so the overall complexity is $O(t_1(t_2 s + dn^2))$. Note that the cost of objective function has not been considered in this complexity analysis since it highly depends on the problem. The pseudocode of the GWOTS is presented in Algorithm 3.

The following remarks can explain why the GWOTS can potentially demonstrate a better efficacy compared to the GWO:

- The GWOTS is able to show a faster convergence rate because it has an enhanced TS-based local search engine and can explore more promising solutions from the vicinity of leaders in each hunting phase.
- The simultaneous use of the unique local searching capacities of TS within the exploration and exploitation phases of the GWO can constructively lead to improved results and performance.
- The utilized TS-based operator can avoid the GWO from cycling drawback. The reason is that it can guide the agents toward some trial positions in the vicinity of leaders, which assists the proposed GWOTS in avoiding the recently visited locations.
- In GWOTS, the leaders are guided to reach to the better local choices around them using the info collected throughout the hunting process.
- The use of TS within the GWO as a local search engine allows the hybrid algorithm to preserve the diversity of wolves and prevent leading to deceptive local solutions.

5 GWOTS for clustering

In this section, the GWOTS is evaluated and applied to the clustering task. The proposed GWOTS-based clustering approach aims to formulate the clustering problem as an optimization problem by following the same concept of the partitional clustering-based algorithms.

The main objective of the GWOTS is to distribute some of the data instances into a number of predefined groups. In the proposed algorithm, the GWOTS tries to locate the optimal

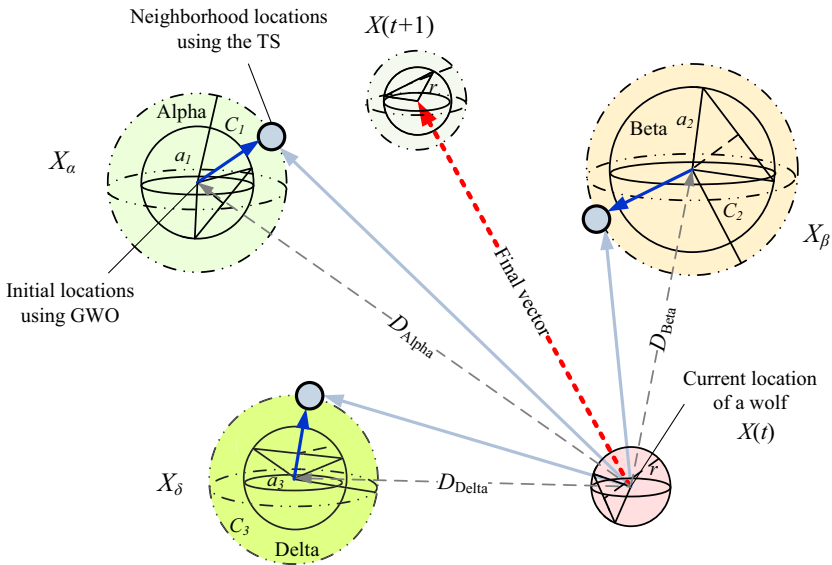


Fig. 3 The 3D motion of wolves in GWOTS

Algorithm 3 Pseudocode of GWOTS

```

Initialize the grey wolf population  $X_i (i = 1, 2, \dots, n)$ 
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of each search agent
 $X_\alpha$  = the best search agent
 $X_\beta$  = the second best search agent
 $X_\delta$  = the third best search agent
while ( $t < Max$  number of iterations) do
  for each leader in ( $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ ) do
    Update the position of the leader agent by Algorithm 2
  end for
  for each search agent do
    Update the position of current search agent by Eq. (5)
  end for
  Update  $a$ ,  $A$ , and  $C$ 
  Calculate the fitness of all search agents
  Update  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ 
   $t = t + 1$ 
end while
return  $X_\alpha$ 

```

▷ Call Algorithm 2

centroid for each group in such a way that each wolf in GWOTS represents a solution that contains the cluster/group of centroids. The following subsections will discuss the proposed GWOTS-based clustering technique in details.

5.1 Clustering preliminaries

Clustering problem can be formulated as follows. Suppose we have a dataset D , which consists of R data points; r_1, r_2, \dots, r_R . Each data point is represented with m -dimensions such as $r_i = (r_{i1}, r_{i2}, \dots, r_{im})$. Given D dataset, a clustering algorithm tries to locate a

set of k clusters such as $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$. Clusters \mathcal{C} are represented by centroids c_j , $j = 1, 2, \dots, k$. In general, the clustering algorithm objective is to maximize the similarity between clusters' members and minimize the similarity between the members from different clusters.

5.2 The proposed GWOTS-based clustering approach

The proposed algorithm is based on GWOTS and consists of four main phases: individual encoding and population initialization, fitness evaluation and update, centroids update, and clustering evaluation. Each of these phases is discussed as follows:

- Individual encoding and population initialization: The method is used to encode the population individuals and form the final clustering solution considering each individual of the population as a complete clustering solution. Each individual of the population represents a vector of centroids with length k , which represents the predefined number of the clusters. Each centroid represents a sub-vector consisting of m dimensions, which reflects the number of the attributes in the dataset D . The length of each individual S is calculated based on the following formula: $S = k \times m$. Figure 4 shows the encoding of the individual.

All population individuals with size N are initialized randomly such that each individual is initialized by S data points, which are selected from the given dataset.

- Fitness evaluation: Similarly to other optimization problems, a fitness function should be defined to assess the clustering quality. As mentioned in the previous subsection, the clustering algorithm tries to minimize and maximize some similarity measures to achieve the best clustering results. In this paper, the most known clustering quality measure called sum of squared errors (SSE) [46,62] is used as the fitness function. The SSE fitness function is calculated based on the Euclidean distances between the data points in the dataset and the nearest centroids in the clusters. The SSE can be calculated using Eq. (8):

$$SSE = \sum_{j=1}^k \sum_{i=1}^{|C_j|} \sigma(c_j, r_i)^2 \tag{8}$$

where σ represents the Euclidean distance between the centroid c_j and i th data point r_i and is given by Eq. (9):

$$\sigma(c_j, r_i) = \sqrt{\sum_{w=1}^m (c_{jw} - r_{iw})^2} \tag{9}$$

where r_{iw} represents w th dimension of the i th data point that belongs to the cluster with c_j centroid.

The fitness function SSE is utilized here to evaluate the goodness of each wolf, such as each wolf tries to minimize its fitness.

- Centroids update: Through the process of GWOTS algorithm, each individual updates its centroids using the best three individuals (α , β , and δ). These three individuals are further enhanced and improved by applying the TS algorithm (Algorithm 2) on each of them independently in an attempt to improve the quality of the leaders. The resulted enhanced leader by the mechanisms of TS will be referred to as (α' , β' , and δ'). Then, each individual moves toward these best individuals (α' , β' , and δ') by updating its centroids.

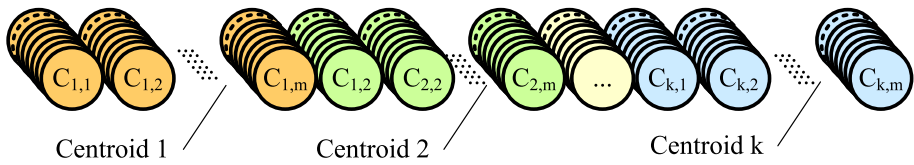


Fig. 4 Individual encoding

The next step is to re-evaluate the SSE of the updated individuals. This iterative process is continued until the maximum number of iterations is achieved. After the clustering process is terminated, the best individual α , which contains the optimal centroids, is utilized to evaluate the clustering results. In this procedure, the TS-based local search exploits the vicinity of the top solutions (α , β , and δ) in each iteration to find any possible better position in the same iteration (see the dotted spheres in Fig. 3). Therefore, the new variant of GWO is locality-informed and neighborhood of each top solution is detected before any further progress in exploration and exploitation trends.

- Clustering evaluation: To assess the clustering quality of the final solution, we used two popular clustering measures: purity and entropy measures [7,63,71]. The purity is the percentage of the number of data points that are clustered correctly. Purity is calculated using Eq. (10):

$$\text{Purity} = \frac{1}{R} \sum_{i=1}^k |L_i \cap C_i| \quad (10)$$

where R represents the number of data points in the dataset; k represents the number of clusters in the dataset; L_i represents the data points that are truly assigned to the actual cluster; and C_i represents the data points that are assigned to cluster i using the clustering algorithm. Entropy measures the semantic distribution of the data points within each cluster, and it is calculated by Eq. (11):

$$\text{Entropy} = \sum_{i=1}^k \frac{|C_i|}{n} E(C_i) \quad (11)$$

where $E(C_i)$ represents the individual entropy of the i th cluster. $E(C_i)$ is given by Eq. (12):

$$E(C_i) = \frac{-1}{\log(k)} \sum_{i=1}^k \frac{|L_i \cap C_i|}{C_i} \log\left(\frac{|L_i \cap C_i|}{C_i}\right) \quad (12)$$

The flow chart of the GWOTS-based clustering algorithm is depicted in Fig. 5. Furthermore, Fig. 6 shows an example of a clustering process on an artificial dataset with 2 attributes.

6 Experimental results and discussion

In this section, the proposed GWOTS is applied to a series of 13 well-studied clustering benchmark tasks with various features and instances, which are obtained from the UCI machine learning repository [15]. Table 1 shows the details of the utilized benchmarks.

In order to further substantiate the exploration and exploitation capacities of the GWOTS, it is compared to several well-recognized evolutionary and swarm-based optimizers such as the

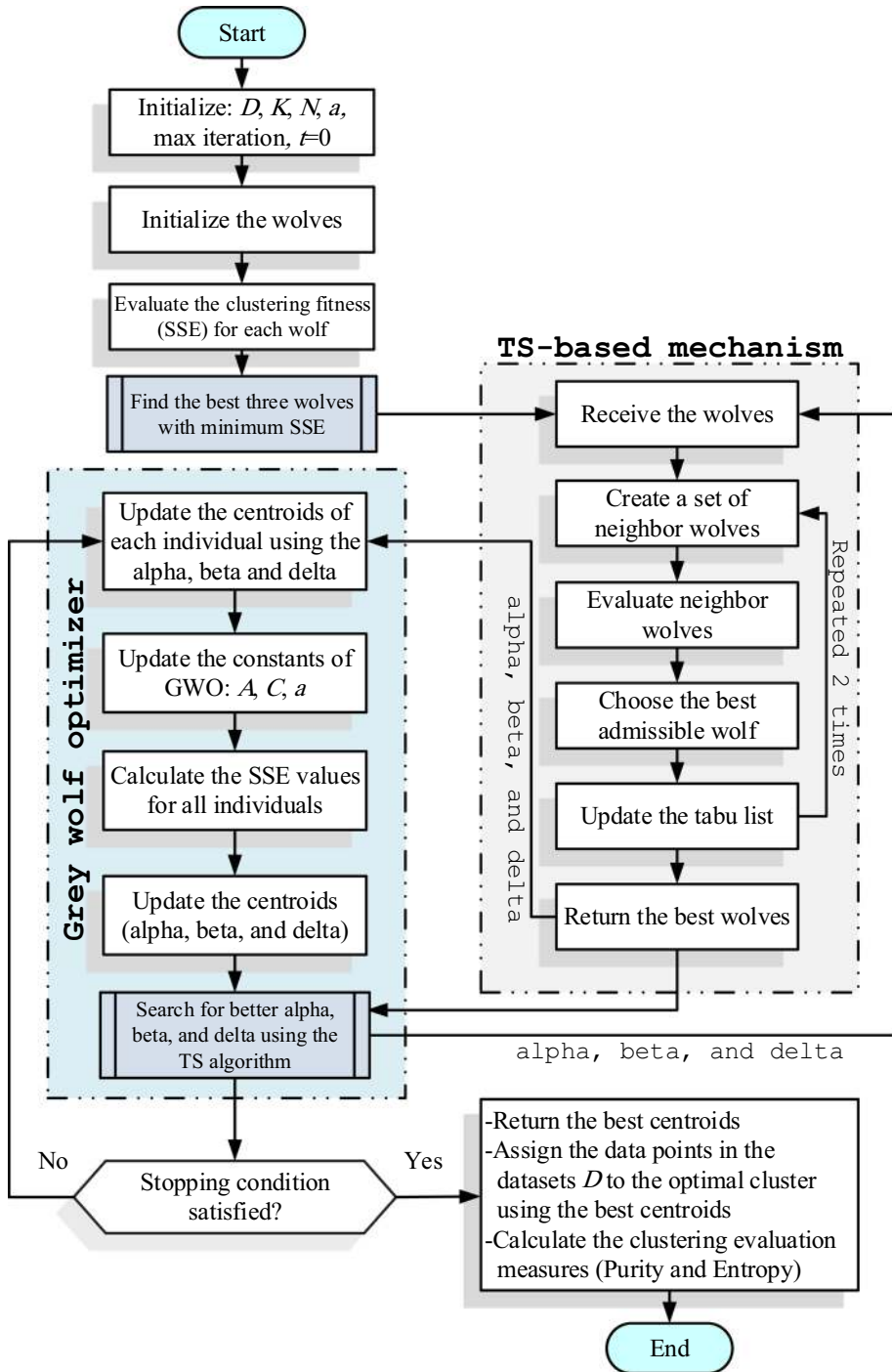


Fig. 5 Flow chart of the GWOTS clustering algorithm

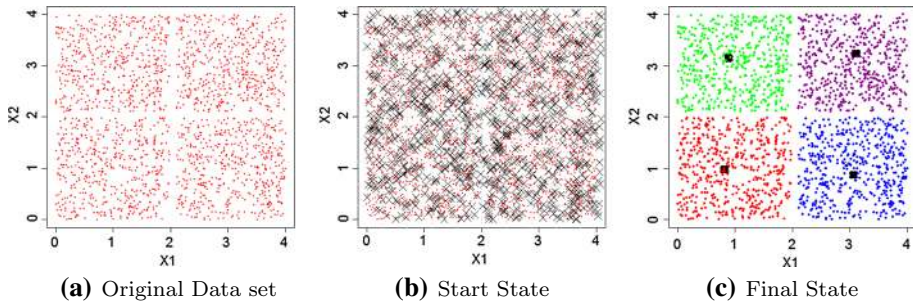


Fig. 6 Clustering process for the artificial dataset, a maximum number of iterations = 200. The individuals start from an initial random data points that are selected from the given dataset and move to the solution (optimal centroids). **a** The original dataset. **b** The initial random individual's locations (small black crosses) with dataset instances (red points). **c** The final solution centroids locations after the clustering process with 4 centroids (small squares), and each cluster in the dataset has a different color based on the minimum distances to the centroid (color figure online)

Table 1 List of utilized datasets

Dataset	Features	Instances	Classes
Iris	4	150	3
Blood	4	778	2
Breast cancer	10	683	2
Glass	9	214	6
Seeds	7	210	3
Wine	13	178	3
Australian	15	653	2
Diabetes	8	768	2
Haberman	3	306	2
Heart	13	270	2
Liver	6	345	2
Planning relax	12	182	2
Tic-tac-toe	9	958	2

TS [23], SA [57], GA [24], PSO [40], DE [70], and original GWO [54] on clustering problems. Several works have utilized these widespread algorithms to substantiate the competences of their methodologies in dealing with clustering test suites, and these approaches revealed efficient performances in handling difficult tasks [12,51,60]. Hence, they also utilized here to validate the efficacy of the proposed GWOTS in handling the clustering tasks. In addition, we compare the results of GWOTS with those computed by commonly used methods such as the well-regarded K -means [36], K -medoids [59], hierarchical clustering (HC) [75], and furthest first (FF) [11] techniques.

All the experiments and comparative evaluations in this research are performed and organized using a PC with Intel Core(TM) i5-2400 3.1GHz CPU and 4.0GB RAM. All tests are implemented under the same fair computational conditions.

The parameters of utilized techniques are tabulated in Table 2. The maximum iterations for SA and TS (trajectory-based algorithms) were set to 10,000, and for others (population-

Table 2 The parameters of algorithms

Algorithm	Parameter	Value
PSO	Inertia factor	0.1
	c_1	2
	c_2	2
FFA	α	0.5
	β	0.2
	γ	1
DE	C Crossover	0.5
	F Scaling	[0.2 0.8]
CS	p_a	0.25
GA	Crossover	0.8
	Mutation	0.02
GWO	a	[2 0]
MFO	a	[-2 -1]
	b	1
TS	bw	0.01
	$Tabu\ size$	5

based optimizers), it was set to 200. The population size was set to 50 for population-based methods. The results are collected over 30 independent runs.

In order to detect the significant improvements of the GWOTS over other algorithms, the results of the Wilcoxon rank sum test are also provided in addition to the mean and standard deviation values. For these tests, the proposed GWOTS for each benchmark is marked as the base method to calculate the p values versus other approaches.

6.1 Comparison with other evolutionary approaches

The results and efficacy of the proposed GWOTS and other algorithms are compared and discussed based on the utilized measures on clustering, namely SSE, purity, and entropy. Table 3 exposes the SSE results of GWOTS and other competitors in realizing all benchmark datasets.

From Table 3, it is observed that the GWOTS can provide the lowest SSE results for 10 datasets: Iris, Blood, Breast, Glass, Seeds, Wine, Australian, Diabetes, Heart, and Tic-tac-toe. It is also the second best metaheuristic on the rest of datasets: Haberman, Liver, and Planning relax. According to STD values, the accuracy of the GWOTS is improved in all cases compared to the basic GWO, especially for Australian, Breast, and Diabetes cases. For Iris, Glass, and Wine, it is seen that the worst SSE results of GWOTS are still better than the best SSE values attained by the GWO to some extent. The median index of GWOTS and TS is similar, but the average results of GWOTS are superior to those of TS on 10 datasets.

Based on the overall ranks, the GWOTS outperforms all others and the TS, GWO, DE, PSO, SA, and GA are in the subsequent places. The GWOTS can demonstrate improved behaviors and searching capabilities and determine relatively preferable results because it can switch from exploration to an extensive TS-based local search more smoothly during the clustering process in comparison with the GWO and other examined metaheuristics. With adding more iterations, it can avoid cycling drawbacks and convergence to local optima.

Table 3 The SSE clustering results for the benchmark datasets

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
GWOTS	MEAN	6.99812	37.96201	205.8811	19.50038	22.02521	48.97581	657.1568	121.25797	25.28983	314.2818	29.98164	59.99926	1183.3973
	STD	0.0000	0.0000	0.00001	0.59011	0.00108	0.01759	6.25583	0.00022	0.03018	5.88341	0.01486	0.20161	1.49347
	MEDIAN	6.99812	37.96201	205.8811	19.4279	22.02438	48.97252	654.391	121.25787	25.28029	312.4192	29.98893	59.8845	1183.16775
	BEST	6.99811	37.96201	205.8811	18.34791	22.02437	48.95405	654.391	121.25785	25.28029	312.4183	29.96446	59.85978	1182.53841
	WORST	6.99812	37.96201	205.8811	20.43859	22.02646	49.00972	673.0279	121.25844	25.37572	331.0263	29.99671	60.29243	1187.51703
	RANK	1	1	1	1	1	1	1	1	2	1	2	2	1
GWO	MEAN	8.39908	38.23718	215.7363	28.78087	24.58126	75.47749	709.0241	133.83055	25.4956	322.2463	31.5537	71.66946	1183.59808
	STD	2.04275	0.44346	25.02934	2.4956	3.74697	9.1783	50.4626	12.56154	0.35192	8.84536	2.30253	2.60004	1.46619
	MEDIAN	7.24064	37.96339	206.7955	28.27197	23.47722	72.1335	690.671	129.60476	25.28122	319.2282	30.05561	73.00308	1183.3422
	BEST	7.00045	37.96232	206.2956	25.47442	22.04294	65.90965	659.275	121.28365	25.28048	313.6094	29.96547	66.09018	1182.66565
	WORST	11.68466	38.93036	286.6496	32.87386	34.82055	95.33539	794.4288	154.76834	26.11308	336.1464	36.0527	73.3419	1187.61685
	RANK	2	3	3	3	3	3	3	3	3	3	3	3	3
TS	MEAN	8.41825	37.96202	205.8811	23.62258	23.30412	50.2861	660.3031	121.2581	25.28029	319.1274	29.97677	59.95393	1186.4965
	STD	2.31122	0.0000	0.00004	3.16713	4.04391	4.17601	7.94724	0.0003	0.0000	8.91002	0.01607	0.17879	5.48168
	MEDIAN	6.99812	37.96201	205.8811	23.60704	22.02544	48.96066	654.3919	121.25793	25.28029	312.434	29.96447	59.8729	1185.43641
	BEST	6.99812	37.96201	205.8811	19.10917	22.02438	48.95419	654.3912	121.25787	25.28029	312.4184	29.96446	59.85981	1182.53853
	WORST	12.14369	37.96202	205.8812	30.88113	34.8133	62.1712	671.0902	121.25856	25.28029	331.0264	29.99764	60.29252	1200.61956
	RANK	3	2	1	2	2	2	2	2	2	2	1	1	3
DE	MEAN	10.08043	43.86099	250.9048	49.24797	38.7596	97.24389	860.0504	163.13824	26.484	366.1035	38.81261	81.19094	1222.64724
	STD	0.84519	5.33365	12.94197	5.87662	4.34617	7.36595	70.50186	15.1056	1.03491	19.14247	3.60689	3.74672	12.03932
	MEDIAN	9.82682	41.55829	248.3042	49.09663	38.07088	95.26384	863.4012	164.70365	26.45121	365.3936	37.45514	80.63802	1223.8916
	BEST	9.12412	38.24826	235.1095	40.06156	30.46827	87.69704	738.0864	141.08226	25.47725	342.5588	35.56858	76.19575	1200.74388
	WORST	11.65915	55.11088	277.2241	60.95043	47.0166	113.156	961.1278	192.33671	29.11746	398.426	45.77838	90.29948	1244.12108
	RANK	4	4	4	5	5	5	4	4	4	5	4	5	5

Table 3 continued

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
PSO	MEAN	10.33011	53.77973	350.2869	46.52517	31.9517	95.39235	863.3661	170.91928	26.33397	376.7482	37.41702	73.1378	1190.08252
	STD	1.74583	12.26652	44.31318	9.06443	5.63711	10.57242	48.10578	14.81124	0.74071	29.49627	4.20291	0.49109	10.64115
	MEDIAN	10.36033	52.26193	355.0491	43.50868	33.02371	95.60125	879.9295	165.4518	26.11286	371.0095	36.63953	73.33181	1187.08107
	BEST	7.30738	39.02247	290.8432	33.36376	22.2405	72.32723	762.6711	157.29688	25.28029	339.234	29.96479	71.78543	1183.09957
	WORST	12.45293	74.17509	414.309	62.26326	42.77746	117.1559	928.2508	203.49327	27.1167	420.3571	45.12322	73.33237	1218.53749
	RANK	5	5	6	4	4	4	5	5	4	5	4	4	4
GA	MEAN	16.87869	74.91705	432.2197	89.23814	55.00826	133.0004	1141.469	249.56716	30.82683	425.6258	55.52705	114.8493	1283.36424
	STD	2.52121	25.29909	85.10503	13.29072	9.31363	19.55916	96.89295	54.22539	2.37167	24.33011	10.64582	11.58467	30.33523
	MEDIAN	16.17139	75.37993	403.492	88.21266	54.66862	135.6601	1099.618	252.78961	30.51188	428.9355	52.8086	116.7077	1292.09945
	BEST	12.49609	49.31944	318.0792	67.03683	43.02129	105.8282	1018.259	177.1857	28.30007	385.1701	43.02903	93.19499	1228.81536
	WORST	20.76783	134.6518	613.668	117.7992	76.12615	171.9004	1273.447	321.0704	36.53907	460.9753	78.04453	126.8765	1325.19866
	RANK	7	7	7	7	6	6	7	7	7	7	7	6	6
SA	MEAN	15.31382	66.3236	257.5605	64.53008	57.8428	137.4646	934.0941	222.88259	30.44052	410.7586	49.29288	116.0231	1299.15363
	STD	2.75615	8.33006	11.21515	15.25839	9.7591	25.64682	52.81378	27.84963	2.68465	27.97174	8.2572	21.26472	36.36612
	MEDIAN	14.7122	66.61275	257.6631	62.32701	55.34884	135.0481	946.5887	226.00736	29.86418	405.0641	48.67414	115.5376	1305.66147
	BEST	11.14464	55.99363	239.0967	42.7511	46.1464	104.6514	839.9915	172.50756	27.20866	377.9663	39.86259	80.47988	1238.4622
	WORST	20.01162	84.71954	277.0955	89.62431	75.22923	190.416	999.2637	276.45597	36.15984	455.8699	63.08775	150.6715	1358.60953
	RANK	6	6	5	6	7	7	6	6	6	6	6	7	7

The best values are marked in bold

Table 4 Purity clustering results of benchmark datasets

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
GWOTS	MEAN	0.88667	0.76203	0.95422	0.54439	0.88857	0.95506	0.83797	0.66810	0.73529	0.77704	0.57971	0.71429	0.65344
	STD	0.00000	0.00000	0.00000	0.00832	0.00246	0.00917	0.03989	0.00041	0.00000	0.06483	0.00000	0.00000	0.00000
	MEDIAN	0.88667	0.76203	0.95422	0.54673	0.89048	0.95506	0.85507	0.66797	0.73529	0.79630	0.57971	0.71429	0.65344
	BEST	0.88667	0.76203	0.95422	0.55140	0.89048	0.96629	0.85507	0.66927	0.73529	0.80000	0.57971	0.71429	0.65344
	WORST	0.88667	0.76203	0.95422	0.52804	0.88571	0.93820	0.73333	0.66797	0.73529	0.59259	0.57971	0.71429	0.65344
	RANK	1	3	2	1	1	1	1	1	1	1	2	2	3
GWO	MEAN	0.82200	0.76203	0.95451	0.49579	0.87048	0.59888	0.75043	0.65742	0.73529	0.76148	0.57971	0.71429	0.65344
	STD	0.10419	0.00000	0.00222	0.01713	0.07785	0.07268	0.12949	0.00738	0.00000	0.04905	0.00000	0.00000	0.00000
	MEDIAN	0.88667	0.76203	0.95565	0.49533	0.90000	0.62360	0.81159	0.65365	0.73529	0.78519	0.57971	0.71429	0.65344
	BEST	0.88667	0.76203	0.95565	0.52336	0.91429	0.64607	0.85507	0.66797	0.73529	0.80000	0.57971	0.71429	0.65344
	WORST	0.66667	0.76203	0.94850	0.46262	0.65714	0.39888	0.55507	0.65104	0.73529	0.65185	0.57971	0.71429	0.65344
	RANK	3	3	1	3	2	3	3	6	1	2	2	2	3
TS	MEAN	0.82067	0.76203	0.95422	0.50654	0.86524	0.92079	0.83333	0.66797	0.73529	0.72852	0.57971	0.71429	0.65344
	STD	0.10627	0.00000	0.00000	0.02673	0.07316	0.10259	0.03822	0.00000	0.00000	0.09732	0.00000	0.00000	0.00000
	MEDIAN	0.88667	0.76203	0.95422	0.50701	0.88810	0.94944	0.85507	0.66797	0.73529	0.79630	0.57971	0.71429	0.65344
	BEST	0.88667	0.76203	0.95422	0.55140	0.89048	0.96629	0.85507	0.66797	0.73529	0.80000	0.57971	0.71429	0.65344
	WORST	0.66667	0.76203	0.95422	0.44860	0.65714	0.62921	0.73333	0.66797	0.73529	0.59259	0.57971	0.71429	0.65344
	RANK	4	3	2	2	3	2	2	2	1	3	2	2	3
DE	MEAN	0.86733	0.76203	0.92718	0.45047	0.77048	0.58202	0.67087	0.66680	0.73529	0.68815	0.57971	0.71429	0.65344
	STD	0.03777	0.00000	0.01976	0.03201	0.09162	0.11971	0.12386	0.01448	0.00000	0.10174	0.00000	0.00000	0.00000
	MEDIAN	0.87333	0.76203	0.92918	0.45093	0.80476	0.60955	0.61087	0.66406	0.73529	0.71111	0.57971	0.71429	0.65344
	BEST	0.93333	0.76203	0.95565	0.48131	0.85238	0.73034	0.86087	0.69010	0.73529	0.80370	0.57971	0.71429	0.65344
	WORST	0.82000	0.76203	0.89843	0.36916	0.62857	0.39888	0.55507	0.65104	0.73529	0.55556	0.57971	0.71429	0.65344
	RANK	2	3	5	5	4	5	5	3	1	7	2	2	3

Table 4 continued

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
PSO	MEAN	0.77133	0.76270	0.90773	0.44206	0.76095	0.41966	0.63913	0.65273	0.73529	0.70148	0.58000	0.71484	0.65344
	STD	0.09270	0.00144	0.02804	0.04295	0.11586	0.06378	0.12099	0.00403	0.00000	0.10612	0.00092	0.00174	0.00000
	MEDIAN	0.77333	0.76203	0.90486	0.45561	0.75000	0.39888	0.55580	0.65104	0.73529	0.74259	0.57971	0.71429	0.65344
	BEST	0.88667	0.76604	0.95851	0.48131	0.88571	0.60112	0.86377	0.66406	0.73529	0.81481	0.58261	0.71978	0.65344
	WORST	0.66667	0.76203	0.85980	0.35514	0.62381	0.39888	0.55507	0.65104	0.73529	0.55556	0.57971	0.71429	0.65344
	RANK	5	1	6	7	6	7	7	7	7	1	6	1	1
GA	MEAN	0.75333	0.76243	0.90587	0.48972	0.76762	0.59888	0.63971	0.65898	0.73529	0.70519	0.57971	0.71429	0.65344
	STD	0.08433	0.00127	0.03512	0.03763	0.08056	0.07718	0.09408	0.01599	0.00000	0.08145	0.00000	0.00000	0.00000
	MEDIAN	0.74000	0.76203	0.91345	0.47430	0.77143	0.61236	0.61014	0.65104	0.73529	0.73148	0.57971	0.71429	0.65344
	BEST	0.92667	0.76604	0.94278	0.56542	0.86667	0.69663	0.83333	0.70052	0.73529	0.80000	0.57971	0.71429	0.65344
	WORST	0.66667	0.76203	0.83834	0.44393	0.64762	0.44944	0.55507	0.65104	0.73529	0.58148	0.57971	0.71429	0.65344
	RANK	6	2	7	4	5	3	6	5	1	1	5	2	3
SA	MEAN	0.75067	0.76203	0.93462	0.44252	0.71476	0.52584	0.67188	0.66211	0.73529	0.70630	0.57971	0.71484	0.65344
	STD	0.09544	0.00000	0.02039	0.04215	0.08445	0.10206	0.10771	0.01934	0.00000	0.09650	0.00000	0.00174	0.00000
	MEDIAN	0.71667	0.76203	0.93634	0.45327	0.70000	0.58427	0.68116	0.65104	0.73529	0.70926	0.57971	0.71429	0.65344
	BEST	0.90000	0.76203	0.95708	0.48131	0.87619	0.62921	0.81739	0.71224	0.73529	0.82222	0.57971	0.71978	0.65344
	WORST	0.66667	0.76203	0.90272	0.36449	0.61429	0.39888	0.55507	0.65104	0.73529	0.56667	0.57971	0.71429	0.65344
	RANK	7	3	4	6	7	6	4	4	1	1	4	2	1

The best values are marked in bold

Table 5 Entropy clustering results of benchmark datasets

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
GWOTS	MEAN	0.26358	0.79142	0.26793	0.56357	0.33019	0.14427	0.60691	0.88531	0.83309	0.74351	0.98154	0.86039	0.92405
	STD	0.00000	0.00000	0.00000	0.01751	0.00463	0.02278	0.09209	0.00032	0.00001	0.07796	0.00006	0.00125	0.00267
	MEDIAN	0.26358	0.79142	0.26793	0.56040	0.32660	0.14471	0.56552	0.88534	0.83309	0.72126	0.98158	0.85997	0.92389
	BEST	0.26358	0.79142	0.26793	0.53241	0.32660	0.11713	0.56552	0.88451	0.83309	0.71299	0.98146	0.85861	0.92032
	WORST	0.26358	0.79142	0.26793	0.59110	0.33557	0.18701	0.83495	0.88582	0.83311	0.96516	0.98158	0.86181	0.92980
	RANK	1	5	2	1	1	1	1	1	4	1	7	4	2
GWO	MEAN	0.32035	0.79146	0.26606	0.62082	0.34362	0.61111	0.72307	0.89463	0.83302	0.76078	0.98122	0.86010	0.92376
	STD	0.09203	0.00008	0.00985	0.03400	0.08287	0.13461	0.17349	0.02946	0.00049	0.05061	0.00040	0.00343	0.00272
	MEDIAN	0.26358	0.79142	0.26092	0.63095	0.31305	0.57427	0.64913	0.88824	0.83309	0.73919	0.98133	0.86141	0.92268
	BEST	0.26358	0.79142	0.26092	0.56312	0.27930	0.51743	0.56552	0.83489	0.83176	0.71299	0.98051	0.85467	0.92126
	WORST	0.46659	0.79157	0.29268	0.66959	0.55760	0.97907	0.99109	0.93309	0.83375	0.86730	0.98158	0.86312	0.92980
	RANK	4	7	1	3	2	3	3	3	3	2	5	2	1
TS	MEAN	0.31529	0.79142	0.26793	0.59599	0.35329	0.18998	0.61948	0.88549	0.83309	0.80528	0.98150	0.86016	0.92651
	STD	0.08419	0.00000	0.00000	0.03803	0.07193	0.13784	0.08680	0.00023	0.00000	0.11911	0.00012	0.00119	0.00356
	MEDIAN	0.26358	0.79142	0.26793	0.58555	0.33108	0.15222	0.56552	0.88534	0.83309	0.72126	0.98158	0.85969	0.92693
	BEST	0.26358	0.79142	0.26793	0.54014	0.32660	0.11713	0.56552	0.88534	0.83309	0.71299	0.98130	0.85861	0.92203
	WORST	0.46659	0.79142	0.26793	0.66323	0.55760	0.58062	0.83495	0.88582	0.83309	0.96516	0.98158	0.86181	0.93003
	RANK	3	5	2	2	3	2	2	2	4	3	6	3	3
DE	MEAN	0.27222	0.79084	0.36466	0.68205	0.44985	0.69294	0.85077	0.90856	0.83313	0.85330	0.97984	0.86128	0.92660
	STD	0.04379	0.00149	0.06626	0.04515	0.10567	0.18513	0.15193	0.01876	0.00055	0.11203	0.00208	0.00205	0.00486
	MEDIAN	0.27664	0.79135	0.36266	0.67478	0.41566	0.59598	0.93437	0.91049	0.83312	0.85073	0.98063	0.86161	0.92743
	BEST	0.21191	0.78663	0.26092	0.63147	0.33300	0.51678	0.55493	0.88095	0.83219	0.70634	0.97471	0.85610	0.91528
	WORST	0.34021	0.79159	0.45524	0.79821	0.63013	0.97426	0.99005	0.93061	0.83375	0.98294	0.98151	0.86312	0.93095
	RANK	2	4	5	5	4	4	4	4	6	7	4	5	4

Table 5 continued

Algorithm	Stat. measure	Iris	Blood	Breast	Glass	Seeds	Wine	Australian	Diabetes	Haberman	Heart	Liver	Planning relax	Tic-tac-toe
PSO	MEAN	0.34250	0.78967	0.42163	0.69849	0.45737	0.94772	0.87896	0.93091	0.83282	0.82174	0.97952	0.86210	0.92660
	STD	0.07052	0.00267	0.08932	0.06725	0.11405	0.12182	0.16205	0.00502	0.00056	0.11015	0.00172	0.00244	0.00297
	MEDIAN	0.34402	0.79084	0.43576	0.67235	0.48304	0.98855	0.97945	0.93264	0.83309	0.79934	0.98005	0.86312	0.92726
	BEST	0.24642	0.78475	0.24373	0.64625	0.32839	0.60128	0.57426	0.91678	0.83176	0.69125	0.97619	0.85559	0.92104
	WORST	0.42062	0.79164	0.56270	0.83655	0.59974	0.98855	0.99123	0.93313	0.83309	0.98793	0.98158	0.86312	0.92965
	RANK	5	2	6	6	5	7	6	7	2	2	4	3	7
GA	MEAN	0.39998	0.78546	0.42582	0.63641	0.45985	0.72193	0.90548	0.92012	0.83214	0.84654	0.97880	0.86157	0.92860
	STD	0.09578	0.00962	0.09566	0.04127	0.06536	0.10238	0.10895	0.01773	0.00250	0.09656	0.00550	0.00171	0.00336
	MEDIAN	0.40526	0.78878	0.41729	0.63960	0.46180	0.73591	0.95682	0.92844	0.83308	0.83261	0.98111	0.86205	0.92988
	BEST	0.19146	0.75963	0.31363	0.56619	0.36536	0.57990	0.64970	0.88060	0.82597	0.70599	0.96391	0.85862	0.92046
	WORST	0.51821	0.79151	0.59458	0.69909	0.57000	0.92031	0.99098	0.93308	0.83376	0.97198	0.98158	0.86311	0.93095
	RANK	7	1	7	4	6	5	7	6	1	6	1	6	7
SA	MEAN	0.36360	0.79045	0.33983	0.70300	0.52086	0.76438	0.85261	0.91185	0.83320	0.83038	0.97883	0.85972	0.92744
	STD	0.07384	0.00125	0.07374	0.06329	0.09839	0.18297	0.11595	0.02613	0.00063	0.11629	0.00345	0.00486	0.00473
	MEDIAN	0.40394	0.79101	0.34123	0.68946	0.54009	0.67757	0.83604	0.92254	0.83336	0.84784	0.98052	0.86301	0.92961
	BEST	0.24595	0.78828	0.25428	0.61750	0.33188	0.54954	0.66592	0.86560	0.83170	0.67510	0.97240	0.84956	0.91630
	WORST	0.42062	0.79164	0.44521	0.81684	0.68006	0.98855	0.99123	0.93313	0.83376	0.98701	0.98159	0.86312	0.93094
	RANK	6	3	4	7	7	6	5	5	7	5	5	2	1

The best values are marked in bold

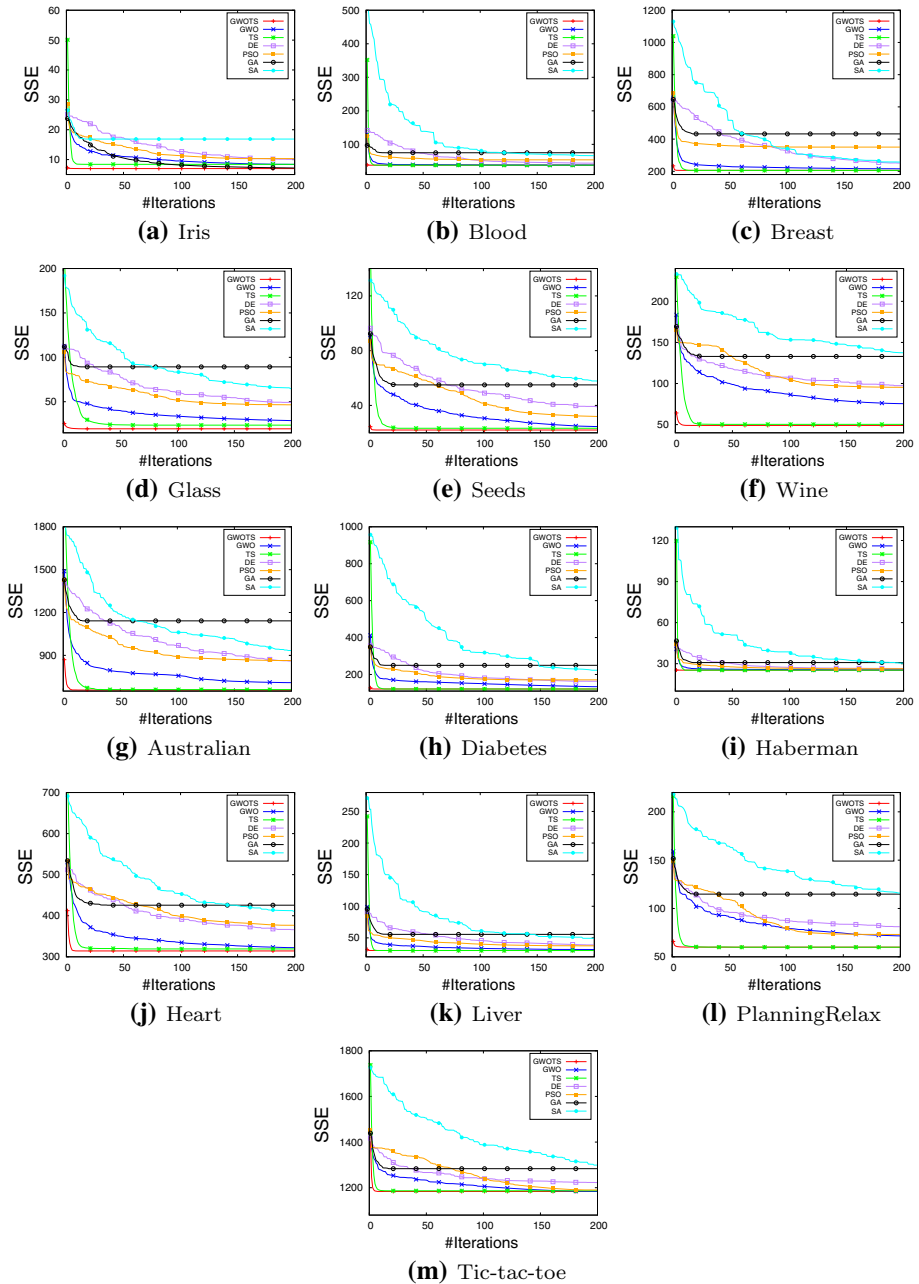


Fig. 7 Convergence curves of the GWOTS versus other methods for all data benchmarks

Table 6 P values of the Wilcoxon test for SSE results of the GWOTS versus other algorithms on all datasets ($p \geq 0.05$ are underlined)

Dataset	TS	GWO	DE	PSO	GA	SA
Iris	9.35E-10	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Blood	5.48E-04	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Breast	2.14E-09	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Glass	2.31E-08	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Seeds	1.15E-04	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Wine	<u>6.95E-01</u>	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Australian	6.14E-06	4.81E-09	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Diabetes	1.14E-05	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Haberman	8.91E-04	1.06E-08	2.87E-11	6.11E-05	2.87E-11	2.87E-11
Heart	2.09E-05	1.06E-08	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Liver	<u>9.00E-01</u>	8.91E-04	2.87E-11	1.06E-08	2.87E-11	2.87E-11
Planning relax	<u>5.99E-01</u>	2.87E-11	2.87E-11	2.87E-11	2.87E-11	2.87E-11
Tic-tac-toe	2.24E-03	2.40E-02	2.87E-11	4.92E-08	2.87E-11	2.87E-11

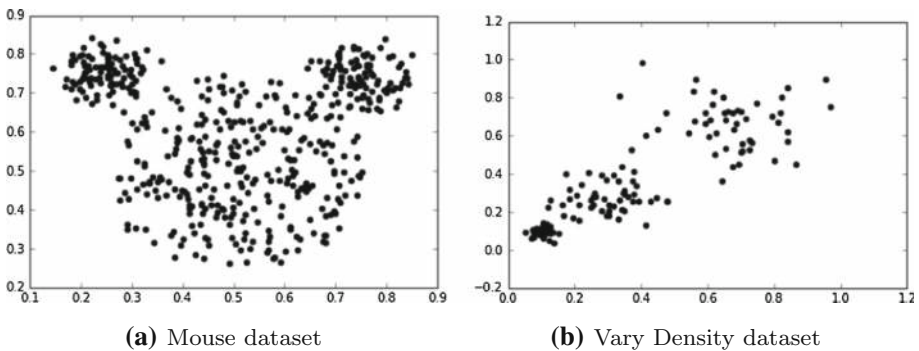
**Fig. 8** Synthetic datasets

Table 7 shows the purity results of different optimizers in dealing with all datasets.

Regarding the purity results in Table 7, the GWOTS can outperform all methods on 7 datasets (Iris, Glass, Seeds, Wine, Australian, Diabetes, and Heart). For Haberman and Tic-tac-toe, all methods have found the same indexes. In the sense of other metrics, several substantial improvements can also be detected in the purity grades. According to the summation of ranks, the GWOTS has classified more objects correctly and the TS, GWO, DE, GA, SA, and PSO have achieved to the subsequent ranks. The TS-based mechanism has effectively increased the chance of GWOTS to avoid the local optima stagnation problem and deepened its exploitative patterns when it wants to exploit the neighborhood of superior leaders during the last steps of the search (Table 4).

Table 5 shows the entropy results of the proposed GWOTS and other algorithms on all 13 datasets.

Inspecting the entropy results in Table 5, it can be seen that once again, the GWOTS shows the highest quality (lowest entropy) compared to the GWO and other competitors. It has enriched the semantic distribution of the data points within each cluster and can

Table 7 Purity clustering results of GWOTS compared with other traditional clustering algorithms

Dataset	Metric	GWOTS	<i>K</i> -means	<i>K</i> -medoids	HC	FF
Iris	Rate	0.88667	0.84267	0.84667	0.88	0.86
	Rank	1	5	4	2	3
Blood	Rate	0.76203	0.76203	0.56551	0.76738	0.76738
	Rank	3	3	5	1	1
Breast	Rate	0.95422	0.95422	0.9113	0.90987	0.83834
	Rank	1	1	3	4	5
Seeds	Rate	0.95506	0.94382	0.8764	0.93258	0.69663
	Rank	1	2	4	3	5
Wine	Rate	0.95506	0.94382	0.8764	0.93258	0.69663
	Rank	1	2	4	3	5
Australian	Rate	0.83797	0.73333	0.67826	0.73333	0.55507
	Rank	1	2	4	2	5
Diabetes	Rate	0.6681	0.66458	0.18099	0.65104	0.65755
	Rank	1	2	5	4	3
Haberman	Rate	0.73529	0.73529	0.52614	0.73529	0.73529
	Rank	1	1	5	1	1
Heart	Rate	0.77704	0.59259	0.58889	0.70741	0.56667
	Rank	1	3	4	2	5
Liver	Rate	0.57971	0.57971	0.26377	0.57971	0.58261
	Rank	2	2	5	2	1
Planning relax	Rate	0.71429	0.71429	0.4011	0.71429	0.71429
	Rank	1	1	5	1	1
Tic-tac-toe	Rate	0.65344	0.65344	0.58664	0.65344	0.65344
	Rank	1	1	5	1	1
Mouse	Rate	0.38878	0.838778	0.86612	0.91	0.8
	Rank	5	3	2	1	4
Vary density	Rate	0.95333	0.86733	0.85333	0.667	0.667
	Rank	1	2	3	4	4
	\sum	21	30	58	31	44
	Overall	1	2	5	3	4

reach the lowest entropy in dealing with 7 datasets. The incorporation of TS as a local enhancement operator permits the GWOTS approach to overlap LO and reveal a satisfactory efficacy. Similar to the efficacies in the sense of purity, all algorithms have competitive entropy indexes in dealing with Blood, Haberman, Liver, Planning relax, and Tic-tac-toe datasets. Regarding the overall ranks, the GWO, TS, and DE are the next superior algorithms. The overall performances also express that PSO, GA, and SA have the same efficacy.

The convergence behaviors are demonstrated and compared in Fig. 7. It is evident that the GWOTS finds promising and superior solutions during the initial steps of the hunting for all datasets. The reason is that the GWOTS can attain better solutions around the leaders in every step, which assist it in maintaining a better balance between the broad exploration and extensive exploitation inclinations as compared to the GWO algorithm.

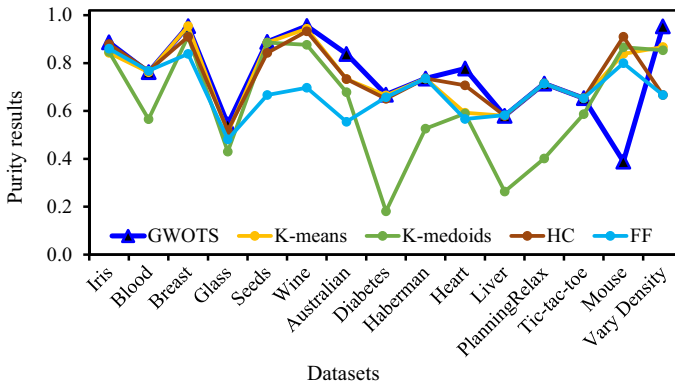


Fig. 9 Visual comparison of purity results

Table 6 presents the statistical test for SSE results of the GWOTS versus other algorithms on all 13 datasets. From Table 6, it is seen that the GWOTS can attain significant improvements over GWO, DE, PSO, GA, and SA optimizers in realizing all datasets. It also provides significantly superior results compared to the TS in the majority of problems.

6.2 Comparison with commonly used clustering techniques

In this section, we compare the results of GWOTS with those obtained by commonly used approaches including the well-known K -means, K -medoids, hierarchical clustering (HC), and furthest first (FF) techniques. These methods have been utilized in various applications as the well-regarded approaches in clustering tasks.

We also include two synthetic datasets to compare the performance of the GWOTS-based clustering approach with the commonly used techniques. The first synthetic dataset is Mouse case, which has 490 records, two features, and three classes. The other case is Vary Density, which includes 150 records, two features, and three classes. These datasets are shown in Fig. 8 as well. These two datasets are generated using ELKI generator and available at (<https://elki-project.github.io/datasets/>).

Tables 5 and 7 expose the purity and entropy results of the GWOTS compared to K -means, K -medoids, HC, and FF techniques.

As per results in Table 7, we see that the proposed GWOTS has obtained the best results on 12 datasets. If we consider the overall ranks, the best method is GWOTS, followed by K -means, HC, FF, and K -medoids techniques, respectively. For Planning relax, Tic-tac-toe, and Haberman cases, we see the GWOTS, FF, HC, and K -means have the same purity index, while for all datasets, the GWOTS is superior to K -medoids technique.

Based on results in Table 5, we see that the proposed GWOTS shows the best efficacy on eight datasets including Breast, Glass, Wine, Australian, Heart, Tic-tac-toe, Mouse, and Vary Density. The second best competitor is K -means, which only obtained the best results on two datasets, and for Breast dataset, its entropy index is the same with GWOTS approach. Based on the overall ranks, we detect that the GWOTS has outperformed all methods in terms of entropy index and the next approaches are K -means, HC, K -medoids, and FF techniques, respectively. The purity and entropy results are also compared visually in Figs. 9 and 10. It is visually observed that the best curve belongs to GWOTS compared to other peers. Table 9

Table 8 Entropy clustering results of GWOTS compared with other traditional clustering algorithms

Dataset	Metric	GWOTS	<i>K</i> -means	<i>K</i> -medoids	HC	FF
Iris	Rate	0.26358	0.30548	0.28607	0.25587	0.30656
	Rank	2	4	3	1	5
Blood	Rate	0.79142	0.79142	0.98431	0.78254	0.78254
	Rank	3	3	5	1	1
Breast	Rate	0.26793	0.26793	0.34782	0.42933	0.59458
	Rank	1	1	3	4	5
Glass	Rate	0.56357	0.59502	0.7502	0.59921	0.64638
	Rank	1	2	5	3	4
Seeds	Rate	0.33019	0.33557	0.32559	0.39118	0.53748
	Rank	2	3	1	4	5
Wine	Rate	0.14427	0.17831	0.3342	0.22074	0.52536
	Rank	1	2	4	3	5
Australian	Rate	0.60691	0.83495	0.90612	0.83473	0.9799
	Rank	1	3	4	2	5
Diabetes	Rate	0.88531	0.56451	0.34861	0.93204	0.92655
	Rank	3	2	1	5	4
Haberman	Rate	0.83309	0.83309	0.99797	0.83361	0.82172
	Rank	2	2	5	4	1
Heart	Rate	0.74351	0.96516	0.97524	0.86958	0.94954
	Rank	1	4	5	2	3
Liver	Rate	0.98154	0.61931	0.40033	0.98153	0.98021
	Rank	5	2	1	4	3
Planning relax	Rate	0.86039	0.5443	0.70646	0.54447	0.86308
	Rank	4	1	3	2	5
Tic-tac-toe	Rate	0.92405	0.92624	0.96571	0.92505	0.93086
	Rank	1	3	5	2	4
Mouse	Rate	0.30153	0.30153	0.374223	0.30165	0.351
	Rank	1	1	5	3	4
Vary density	Rate	0.14481	0.22755	0.29313	0.421	0.466
	Rank	1	2	3	4	5
	\sum	29	35	53	44	59
	Overall	1	2	4	3	5

shows the CPU time spent by GWOTS compared to other peers. As per results, the GWOTS is fast enough and time results are acceptable and expected, but to be fair, all traditional methods are faster than GWOTS. The main reason is that we proposed a swarm-based approach and the nature of operators and the new TS-based mechanism are time-consuming and have their side effect, despite the observed improvements in the clustering results. However, the GWOTS can solve these datasets in a very reasonable time with better quality indexes for most of the cases.

Based on these results, we conclude that the efficacy of the GWOTS is very competitive and better than commonly used methods in most of the cases. The main reason is that the

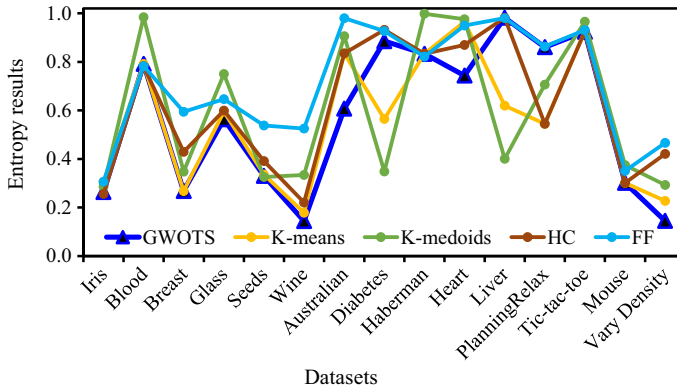


Fig. 10 Visual comparison of entropy results

Table 9 Elapsed time results for GWOTS versus well-known clustering algorithms

Dataset	GWO-TS	<i>K</i> -means	<i>K</i> -medoids	HC	FF
Iris	20.26	2	2	2	3
Blood	68.95	3	3	11	3
Breast	71.37	3	3	3	2
Glass	40.05	3	3	2	2
Seeds	25.64	3	3	2	3
Wine	25.74	3	3	2	3
Australian	80.66	3	3	3	2
Diabetes	78.76	2	2	10	2
Haberman	28.58	2	2	3	2
Heart	33.80	3	3	4	3
Liver	36.23	3	3	5	3
Planning relax	19.17	3	3	2	2
Tic-tac-toe	123.88	11	11	6	3
Mouse	60.9911	3	3	3	3
Vary density	16.2686	2	2	2	2

GWOTS can establish a fine balance between main exploratory and exploitative trends and in the case of finding any high-quality solution; it is locality-informed and can intensify the exploitation around that position, which leads to enhancing the level of quality.

6.3 Discussions

Taken together, the results supported the hypothesis of the paper. It was confirmed that hybridizing GWO and TS improves the performance of GWO, significantly. A wide range of case studies with different difficulties showed that this improvement is beneficial. It can be concluded from the extensive results and comparisons that the proposed GWOTS-based clustering approach has an improved efficacy in terms of SSE, purity, and entropy on different

clustering datasets. Therefore, it can be considered as an capable clustering method with several advantages inherited from both GWO and TS algorithms.

Compared to other swarm- and trajectory-based methods, GWOTS shows a very competitive and superior performance and provides faster convergence behaviors, and it makes a more stable balance between exploration and exploitation trends. The GWOTS inherits all advantages of conventional GWO compared to other swarm-based methods such as dynamic searching mechanisms. It also demonstrates more enhanced exploitative capacities compared to the GA, DE, and PSO due to the effective role of TS-based local search mechanism. Compared to the TS and SA, it has extensive exploration potentials that can assist GWOTS in avoiding local optima and stagnation shortcomings. Compared to well-known traditional methods, it provides very competitive and high-quality results. However, we observed the computational time of the GWOTS is higher than GWO as a side effect of the TS operator. Hence, it is obvious that the GWOTS method needs more computational time compared to the traditional methods for obtaining the optimum clusters, while well-known methods such as K -means and FF should be faster. We proposed a swarm-based method and compared it with some swarm-based and trajectory-based evolutionary methods. The GWOTS method is better than GWO and similar optimizers from the same family with comparable exploration and exploitation operators. We did not intend to claim that GWOTS is the best method in the world because based on no free lunch (NFL) theorem for search and optimization [76], such a method does not exist. There is a rich literature for swarm-based optimizers for clustering, and this is obvious and widely accepted that the traditional methods such as K -means are often faster than swarm-based methods. However, it does not mean that methods such as K -means are also better in terms of the accuracy of the results and local solutions avoidance. In addition, it is evident that when adding TS, the run time of GWO will increase, but in turn, we will get more accurate results. In addition, local search methods often increase the time of the algorithm and they are time-consuming, especially methods like TS. Run time is just one measure of comparison. We have used the accuracy in this word since our main contribution is to improve the accuracy and not the speed. This holds for the majority of swarm-based clustering methods as well. However, there are many techniques to improve the run-time performance of swarm-based methods such as parallel computing, which can be used to speed up the GWOTS-based method as well.

7 Conclusions and future directions

In this paper, an improved GWO-based optimizer was proposed to deal with clustering applications. In GWOTS, an effective TS-based strategy was employed to further search in the proximity of the best solutions obtained so far and improve the performance of GWO. Thirteen clustering datasets were utilized besides ranksum statistical test to assess the efficiency of GWOTS in comparison with former algorithms. The comprehensive results and analysis disclosed the superiority of the GWOTS in terms of optimality of the results and convergence behaviors in dealing with clustering datasets.

The future studies can utilize the proposed GWOTS for tackling other class clustering problems. There are many spatial applications in the field of location-based services (LBS) that the proposed GWOTS-based clustering approach can also be evaluated. In future works, we will investigate the performance of swarm-based and evolutionary clustering methods such as GWOTS on synthetic datasets with different sizes and arbitrary shapes. We will also utilize parallel computing to reduce the run time of the proposed GWOTS method.

References

1. Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. *Energy Convers Manag* 179:362–372
2. Agustí LE, Salcedo-Sanz S, Jiménez-Fernández S, Carro-Calvo L, Del Ser J, Portilla-Figueras JA et al (2012) A new grouping genetic algorithm for clustering problems. *Expert Syst Appl* 39(10):9695–9703
3. Ahmadyfard A, Modares H (2008) Combining pso and k -means to enhance data clustering. In: *IEEE International Symposium on Telecommunications*, 2008, pp 688–691
4. Al-Madi N, Aljarah I, Ludwig SA (2014) Parallel glow worm swarm optimization clustering algorithm based on mapreduce. In: *IEEE Symposium on Swarm intelligence (SIS)*, 2014, pp 1–8
5. Aljarah I, Ludwig SA (2012) Parallel particle swarm optimization clustering algorithm based on mapreduce methodology. In: *IEEE Fourth world congress on nature and biologically inspired computing (NaBIC)*, 2012, pp 104–111
6. Aljarah I, Ludwig SA (2013) Mapreduce intrusion detection system based on a particle swarm optimization clustering algorithm. In: *IEEE congress on evolutionary computation (CEC)*, 2013, pp 955–962
7. Aljarah I, Ludwig SA (2013) A new clustering approach based on glowworm swarm optimization. In: *IEEE congress on evolutionary computation (CEC)*, 2013, pp 2642–2649
8. Aljarah I, Ludwig SA (2013) Towards a scalable intrusion detection system based on parallel pso clustering using mapreduce. In: *Proceedings of the 15th annual conference companion on genetic and evolutionary computation*, ACM, pp 169–170
9. Aljarah I, Mafarja M, Heidari AA, Faris H, Mirjalili S (2020) Multi-verse optimizer: theory, literature review, and application in data clustering. Springer, Cham, pp 123–141
10. Ibrahim A, Majdi M, Asghar HA, Hossam F, Yong Z, Seyedali M (2018) Asynchronous accelerating multi-leader salp chains for feature selection. *Appl Soft Comput* 71:964–979
11. Apiletti D, Baralis E, Bruno G, Cerquitelli T (2009) Real-time analysis of physiological data to support medical applications. *IEEE Trans Inf Technol Biomed* 13(3):313–321
12. Das S, Abraham A, Konar A (2008) Automatic clustering using an improved differential evolution algorithm. *IEEE Trans Syst Man Cybern A Syst Hum* 38(1):218–237
13. Ding Y, Xian F (2016) Kernel-based fuzzy c -means clustering algorithm based on genetic algorithm. *Neurocomputing* 188:233–238
14. Doval D, Mancoridis S, Mitchell BS (1999) Automatic clustering of software systems using a genetic algorithm. In: *STEP'99 proceedings software technology and engineering practice*, IEEE, pp 73–81
15. Dua D, Graff C (2019) UCI machine learning repository. School of Information and Computer Science, University of California. Irvine, CA. <http://archive.ics.uci.edu/ml>
16. Muhammad F, Farhan A, Salabat K, Azmat SP, Khan M, Jaime L, Haoxiang W, Weon LJ, Irfan M et al (2018) Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks. *Comput Electr Eng* 70:853–870
17. Hossam F, Al-Zoubi AM, Asghar HA, Ibrahim A, Majdi M, Hassonah Mohammad A, Hamido F (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Inf Fusion* 48:67–83
18. Faris H, Aljarah I, Mirjalili S, Castillo PA, Merelo JJ (2016) Evolopy: an open-source nature-inspired optimization framework in python. In: *Proceedings of the 8th international joint conference on computational intelligence, IJCCI 2016*, vol 1. ECTA, Porto, Portugal, 9–11 Nov 2016, pp 171–177
19. Hossam F, Mafarja Majdi M, Asghar HA, Ibrahim A, Al-Zoubi AM, Seyedali M, Hamido F (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl Based Syst* 154:43–67
20. Faris H, Mirjalili S, Aljarah I, Mafarja M, Heidari AA (2020) Salp Swarm algorithm: theory, literature review, and application in extreme learning machines. Springer, Cham, pp 185–199
21. Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Comput Oper Res* 13(5):533–549
22. Glover F (1989) Tabu search part i. *ORSA J Comput* 1(3):190–206
23. Glover F, Laguna M (2013) Tabu search. In: *Pardalos PM, Du D-Z, Graham RL (eds) Handbook of combinatorial optimization*. Springer, Boston, pp 3261–3362
24. Goldberg David E (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
25. Gyamfi KS, Brusey J, Hunt A (2017) K -means clustering using Tabu search with quantized means. arXiv preprint [arXiv:1703.08440](https://arxiv.org/abs/1703.08440)
26. Hassanzadeh T, Meybodi MR (2012) A new hybrid approach for data clustering using firefly algorithm and k -means. In: *IEEE 16th CSI international symposium on artificial intelligence and signal processing (AISP)*, pp 007–011

27. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. *Inf Sci* 222:175–184
28. Asghar HA, Ali AR, Rezaee JA (2017) Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems. *Appl Soft Comput* 57:657–671
29. Heidari AA, Aljarah I, Faris H, Chen H, Luo J, Mirjalili S (2019) An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04015-0>
30. Heidari AA, Faris H, Aljarah I, Mirjalili S (2018) An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Comput*. <https://doi.org/10.1007/s00500-018-3424-2>
31. Heidari AA, Faris H, Mirjalili S, Aljarah I, Mafarja M (2020) Ant Lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. Springer, Cham, pp 23–46
32. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
33. Asghar HA, Parham P (2017) An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Appl Soft Comput* 60:115–134
34. Jain Anil K, Narasimha MM, Flynn Patrick J (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264–323
35. Jayabarathi T, Raghunathan T, Adarsh BR, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. *Energy* 111:630–641
36. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k -means clustering algorithm: analysis and implementation. *IEEE Trans Pattern Anal Mach Intell* 7:881–892
37. Kapoor S, Zeya I, Singhal C, Nanda SJ (2017) A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation. *Proc Comput Sci* 115:415–422
38. Katagiri H, Hayashida T, Nishizaki I, Guo Q (2012) A hybrid algorithm based on tabu search and ant colony optimization for k -minimum spanning tree problems. *Expert Syst Appl* 39(5):5681–5686
39. Katarya R, Verma OP (2018) Recommender system with grey wolf optimizer and fcm. *Neural Comput Appl* 30(5):1679–1687
40. Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: *IEEE international conference on evolutionary computation*, pp 303–308
41. Kennedy J (2011) Particle swarm optimization. In: Sammut C, Webb GI (eds) *Encyclopedia of machine learning*. Springer, Berlin, pp 760–766
42. Khairuzzaman AKM, Chaudhury S (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert Syst Appl* 86:64–76
43. Kirkpatrick S, Gelatt CD, Vecchi MP et al (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
44. Korayem L, Khorsid M, Kassem SS (2015) Using grey wolf algorithm to solve the capacitated vehicle routing problem. In: *IOP conference series: materials science and engineering*, IOP Publishing, vol 83, p 012014
45. Kumar V, Chhabra JK, Kumar D (2017) Grey wolf algorithm-based clustering technique. *J Intell Syst* 26(1):153–168
46. Kwedlo W (2011) A clustering method combining differential evolution with the k -means algorithm. *Pattern Recognit Lett* 32(12):1613–1621
47. Lee C-Y, Antonsson EK (2000) Dynamic partitionial clustering using evolution strategies. In: 26th annual conference of the IEEE industrial electronics society, IECON, vol 4, pp 2716–2721
48. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl Based Syst* 161:185–204
49. Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, AlaM A-Z, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl Based Syst* 145:25–45
50. Mafarja M, Heidari AA, Faris H, Mirjalili S, Aljarah I (2020) Dragonfly algorithm: theory, literature review, and application in feature selection. Springer, Cham, pp 47–67
51. Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. *Pattern Recognit* 33(9):1455–1465
52. Mirjalili S (2015) How effective is the grey wolf optimizer in training multi-layer perceptrons. *Appl Intell* 43(1):150–161
53. Mirjalili S, Aljarah I, Mafarja M, Heidari AA, Faris H (2020) Grey Wolf optimizer: theory, literature review, and application in computational fluid dynamics problems. Springer, Cham, pp 87–105
54. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
55. Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitionial clustering. *Swarm Evolut Comput* 16:1–18

56. Omran M, Engelbrecht AP, Salman A (2005) Particle swarm optimization method for image clustering. *Int J Pattern Recognit Artif Intell* 19(03):297–321
57. Osman IH, Christofides N (1994) Capacitated clustering problems by hybrid simulated annealing and tabu search. *Int Trans Oper Res* 1(3):317–336
58. Ozturk C, Hancer E, Karaboga D (2015) Dynamic clustering with improved binary artificial bee colony algorithm. *Appl Soft Comput* 28:69–80
59. Park H-S, Jun C-H (2009) A simple and fast algorithm for k -medoids clustering. *Expert Syst Appl* 36(2):3336–3341
60. Rana S, Jasola S, Kumar R (2011) A review on particle swarm optimization algorithms and their applications to data clustering. *Artif Intell Rev* 35(3):211–222
61. Rao AS, Ramakrishna S, Chitti Babu P (2016) Modc. multi-objective distance based optimal document clustering by ga. *Indian J Sci Technol* 9:1–8
62. Rokach L, Maimon O (2005) Clustering methods. In: Maimon O, Rokach L (eds) *Data mining and knowledge discovery handbook*. Springer, Boston, pp 321–352
63. Rosenberg A, Hirschberg J (2007) V-measure: a conditional entropy-based external cluster evaluation measure. *EMNLP-CoNLL* 7:410–420
64. Scheunders P (1997) A genetic c -means clustering algorithm applied to color image quantization. *Pattern Recognit* 30(6):859–866
65. Senthilnath J, Omkar SN, Mani V (2011) Clustering using firefly algorithm: performance study. *Swarm Evolut Comput* 1(3):164–171
66. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony approach for clustering. *Anal Chim Acta* 509(2):187–195
67. Shen Q, Shi W-M, Kong W (2008) Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Comput Biol Chem* 32(1):53–60
68. Shukri S, Faris H, Aljarah I, Mirjalili S, Abraham A (2018) Evolutionary static and dynamic clustering algorithms based on multi-verse optimizer. *Eng Appl Artif Intell* 72:54–66
69. Song HM, Sulaiman MH, Mohamed MR (2014) An application of grey wolf optimizer for solving combined economic emission dispatch problems. *Int Rev Model Simul* 7(5):838–844
70. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
71. Strehl A, Ghosh J, Mooney R (2000) Impact of similarity measures on web-page clustering. In: *Workshop on artificial intelligence for web search (AAAI 2000)*, vol 58, p 64
72. Talbi E-G (2009) *Metaheuristics: from design to implementation*, vol 74. Wiley, New York
73. Kumar TA, Kapil S, Manju B (2018) A novel clustering method using enhanced grey wolf optimizer and mapreduce. *Big Data Res* 14:93–100
74. Van der Merwe DW, Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: *IEEE congress on evolutionary computation, CEC'03*, vol 1, pp 215–220
75. Wang J, Li M, Chen J, Pan Y (2011) A fast hierarchical clustering algorithm for functional modules discovery in protein interaction networks. *IEEE/ACM Trans Comput Biol Bioinf* 8(3):607–620
76. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evolut Comput* 1(1):67–82
77. Rui X, Wunsch D (2005) Survey of clustering algorithms. *IEEE Trans Neural Netw* 16(3):645–678
78. Zhang S, Zhou Y (2015) Grey wolf optimizer based on powell local optimization method for clustering analysis. *Discrete Dyn Nat Soc* 2015:481360



Ibrahim Aljarah is an associate professor of Big Data Mining and Computational Intelligence at the University of Jordan—Department of Business Information Technology, Jordan. He obtained his bachelor degree in computer science from Yarmouk University—Jordan, 2003. Dr. Aljarah also obtained his master degree in computer science and information systems from the Jordan University of Science and Technology—Jordan in 2006. He also obtained his PhD in computer science from the North Dakota State University (NDSU), USA, in May 2014. He organized and participated in many conferences in the field of data mining, machine learning, and Big Data such as NTIT, CSIT, IEEE NABIC, CASON, and BIGDATA Congress. Furthermore, he contributed in many projects in USA such as Vehicle Class Detection System (VCDS), Pavement Analysis Via Vehicle Electronic Telemetry (PAVVET), and Farm Cloud Storage System (CSS) projects. He has published more than 35 papers in refereed international conferences and journals. His research focuses on data mining, machine learning,

Big Data, MapReduce, Hadoop, swarm intelligence, evolutionary computation, social network analysis (SNA), and large-scale distributed algorithms.



Majdi Mafarja received his B.Sc in software engineering and M.Sc in computer information systems from Philadelphia University and the Arab Academy for Banking and Financial Sciences, Jordan, in 2005 and 2007, respectively. Dr. Mafarja did his PhD in computer science at National University of Malaysia (UKM). He was a member in Data Mining and Optimization Research Group (DMO). Now he is an assistant professor at the Department of Computer Science at Birzeit University. His research interests include evolutionary computation, metaheuristics, and data mining.



Ali Asghar Heidari is now a PhD research intern at the School of Computing, National University of Singapore (NUS). Currently, he is also an exceptionally talented PhD candidate at the University of Tehran and he is awarded and funded by Iran's National Elites Foundation (INEF). His main research interests are advanced machine learning, evolutionary computation, metaheuristics, prediction, information systems, and spatial modeling. He has published more than ten papers in international journals such as Information Fusion, Energy Conversion and Management, Applied Soft Computing, and Knowledge-Based Systems.



Hossam Faris is an associate professor at Business Information Technology Department, King Abdullah II School for Information Technology, the University of Jordan (Jordan). Hossam Faris received his BA, M.Sc. degrees (with excellent rates) in computer science from Yarmouk University and Al-Balqa Applied University in 2004 and 2008, respectively, in Jordan. Since then, he has been awarded a full-time competition-based PhD scholarship from the Italian Ministry of Education and Research to peruse his PhD degrees in e-Business at University of Salento, Italy, where he obtained his PhD degree in 2011. In 2016, he worked as a postdoctoral researcher with GeNeura Team at the Information and Communication Technologies Research Center (CITIC), University of Granada (Spain). His research interests include: applied computational intelligence, evolutionary computation, knowledge systems, data mining, semantic web and ontologies.



Seyedali Mirjalili is a lecturer at Griffith University and internationally recognized for his advances in swarm intelligence (SI) and optimization, including the first set of SI techniques from a synthetic intelligence standpoint—a radical departure from how natural systems are typically understood—and a systematic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimization algorithms. Dr Mirjalili has published over 80 journal articles, many in high-impact journals with over 7000 citations in total with an H-index of 29 and G-index of 84. From Google Scholar metrics, he is globally the third most cited researcher in engineering optimization and robust optimization. He is serving as an associate editor of *Advances in Engineering Software* and the *Journal of Algorithms*.