

Article

# An Information Entropy-Based Animal Migration Optimization Algorithm for Data Clustering

Lei Hou <sup>1</sup>, Jian Gao <sup>1,2,\*</sup> and Rong Chen <sup>1,\*</sup>

<sup>1</sup> College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; gz361301@dlmu.edu.cn

<sup>2</sup> Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

\* Correspondence: gaojian@dlmu.edu.cn (J.G.); rchen@dlmu.edu.cn (R.C.); Tel.: +86-411-8472-3669 (R.C.)

Academic Editor: Raúl Alcaraz Martínez

Received: 16 March 2016; Accepted: 10 May 2016; Published: 16 May 2016

**Abstract:** Data clustering is useful in a wide range of application areas. The Animal Migration Optimization (AMO) algorithm is one of the recently introduced swarm-based algorithms, which has demonstrated good performances for solving numeric optimization problems. In this paper, we presented a modified AMO algorithm with an entropy-based heuristic strategy for data clustering. The main contribution is that we calculate the information entropy of each attribute for a given data set and propose an adaptive strategy that can automatically balance convergence speed and global search efforts according to its entropy in both migration and updating steps. A series of well-known benchmark clustering problems are employed to evaluate the performance of our approach. We compare experimental results with *k*-means, Artificial Bee Colony (ABC), AMO, and the state-of-the-art algorithms for clustering and show that the proposed AMO algorithm generally performs better than the compared algorithms on the considered clustering problems.

**Keywords:** animal migration optimization; information entropy; data clustering

## 1. Introduction

The clustering problem is a basic research topic in data mining [1–3]; it is encountered in a number of academic and practical fields such as text document analysis, web data analysis, image processing, data compression, and bioinformatics. In recent years, we have viewed an increasing number of publications on the models and algorithms of data clustering [4–7], since the topic plays an important role in these fields. The task of clustering is to recognize natural groupings in multidimensional data based on certain similarity measures. For example, Euclidean distance is a measurement for evaluating similarities between clusters, which is one of the most frequently used distances in clustering problems. Specifically, given *N* objects, one should allocate each object to one of *K* clusters with the aim of minimizing the sum of squared Euclidean distances between each object and its corresponding centroid of the cluster. Formally, the problem can be described as follows [8,9]:

$$\text{Min } C(x, z), C(x, z) = \sum_{i=1}^N \sum_{j=1}^K w_{ij} \|x_i - z_j\|^2, i = 1, 2, \dots, N, j = 1, 2, \dots, K, \quad (1)$$

where *N* is the number of patterns, and *K* is the number of clusters;  $x_i$  is the location of the *i*-th pattern, and  $z_j$  is the center of the *j*-th cluster, and it is obtained by the following Equation (2):

$$z_j = \frac{1}{N_j} \sum_{i=1}^N w_{ij} x_i, \quad (2)$$

where  $N_j$  is the number of patterns in the  $j$ -th cluster, and  $w_{ij}$  is the association weight of pattern  $x_i$  to cluster  $j$ , i.e.,  $w_{ij}$  is 1 if pattern  $i$  belongs to cluster  $j$  and 0 otherwise.

Generally, the clustering problem is computationally difficult, namely NP-hard [10], so investigating efficient optimization algorithms to find better clusters is still an important task. The difficulty for designing and improving such algorithms is to propose effective strategies for optimization and to find suitable values of control parameters. Over the last several decades, a wide variety of algorithms and improvements have been presented and analyzed, which are mainly divided in two kinds: hierarchical methods and partitional methods. The  $k$ -means algorithm [11] is a basic and well-known partitional method. It starts from  $k$  random positions (centroids) and iterates by updating these centroids until they are no longer moved. The algorithm aims to minimize an objective function, which can be described as follows:

$$f(X, C) = \sum_{i=1}^N \min \left\{ \|x_i - c_k\|^2 \mid k = 1, 2, \dots, K \right\}, \quad (3)$$

where  $x_i$  is the location of the  $i$ -th pattern, and  $c_k$  is the  $k$ -th centroid. This method is easy to handle, but it often converges to a local optimum, so the quality of results highly depends on the initialization positions.

In order to overcome this issue, an alternative approach is swarm-based or metaheuristic-based optimization algorithms, where the genetic algorithm [12–14], particle swarm optimization [15–17], and the Artificial Bee Colony (ABC) algorithm [18,19] are typical ones. Very recently, some new metaheuristic algorithms have been proposed, such as Monarch Butterfly Optimization (MBO) [20], Elephant Herding Optimization (EHO) [21], and Animal Migration Optimization algorithm (AMO) [22]. Among these, the AMO algorithm, proposed by Li *et al.*, is an efficient one [22]. Recent studies have shown that the AMO algorithm is good at solving many numeric optimization problems, and it performs well on benchmark instances. These metaheuristic-based algorithms have shown excellent performance on solving many optimization problems [23–26]. They can always achieve good solutions compared to other heuristic algorithms. Metaheuristic-based algorithms have also been employed to deal with clustering problems [27–32], since such problems are naturally optimization problems.

In addition, information entropy is a good measure for data clustering, and it is often used as a heuristic. To cluster high-dimensional objects in subspaces and determine the importance of each dimension, Jing *et al.* [33] proposed a method that combines the weight entropy into the objective function to be minimized, and they also introduced an extra step to compute the contribution of each dimension to each cluster. Furthermore, information entropy was used for determining the optimal number of clusters. Liang *et al.* [34] proposed an approach to measure within-cluster entropy and between-cluster entropy with the aim of determining the number of clusters in a given data set effectively. Cheung and Jia [35] investigated clustering on mixed data composed of numerical and categorical attributes, and proposed an iterative clustering algorithm. To analyze similarities between objects, they provided a method to estimate the significance of categorical attributes using information theory, and then showed that the algorithm can determine the number of clusters automatically without prefixing control parameters.

Though there are several clustering algorithms that employ information entropy to analyze similarities of clusters, there are rare algorithms that use entropy as heuristic information for optimization. In this paper, we propose an information entropy-based AMO algorithm for solving clustering problems. The key feature of the algorithm is that a new migration method as well as a new population updating method controlled by information entropy is proposed. The original migration method is designed for common optimization problems, but the clustering data possess their own distribution, and values of attributes always have cluster properties, so the new migration method employs entropy heuristics to control the searching direction of each attribute, and a similar strategy is also incorporated into our population updating method. We perform intensive experiments on benchmark instances. Experimental results show that our new approach can find better clustering

results compared to other approaches presented recently, and further indicate that our new approach accelerates convergence speed of optimization.

The remainder of this paper is organized as follows: In Section 2, we briefly review the AMO algorithm, and the entropy-based AMO algorithm for clustering problems is introduced in Section 3. Benchmarks for evaluating algorithms and experimental results are given in Section 4. Finally, conclusions are provided in Section 5.

## 2. The AMO Algorithm

In this section, we briefly introduce the AMO algorithm. The AMO algorithm is a swarm-based algorithm inspired by the migration phenomenon of animals. In the algorithm, individuals are regarded as positions of animals, and positions can be moved by mainly two operations: animal migration and population updating. The operation of animal migration simulates behaviors of animal groups moving from the current area to a new area. New positions of individuals will be produced according to the direction of animal migration, where three migration rules are considered: Individuals move towards the same direction as their neighboring individuals; individuals remain near their neighboring individuals; and individuals avoid collisions with their neighboring individuals. Using the three migration rules, a probability approach is introduced to yield new positions of individuals. The algorithm begins with a randomly initialization population, which is comprised of  $NP$  feature vectors with  $D_x$  dimensions, which can be stated as follows:

$$x_{i,j,0} = x_{j,min} + rand_{i,j} \cdot (x_{j,max} - x_{j,min}), \tag{4}$$

where  $x_{j,max}$  and  $x_{j,min}$  are the maximum value and the minimum value of the  $j$ -th dimension.  $x_{i,j,0}$  is the  $j$ -th dimension value of the  $i$ -th individual in the initialization population, and  $rand_{i,j}$  is a uniformly random number between 0 and 1,  $i = 1, \dots, NP$  and  $j = 1, \dots, D_x$ .

After producing the initialization population, animal migration and population updating operations are performed iteratively. During the animal migration, the individuals are supposed to move to new positions according to the positions of their neighboring individuals, which can be described as follows:

$$x_{i,j,G+1} = x_{i,j,G} + \delta \cdot (x_{neighborhood,j,G} - x_{i,j,G}), \tag{5}$$

where  $x_{i,j,G}$  is the  $j$ -th dimension value of the  $i$ -th individual in the current population  $G$ , and  $x_{i,j,G+1}$  is the  $j$ -th dimension value of the  $i$ -th individual in the new population  $G + 1$ ;  $x_{neighborhood,j,G}$  is the  $j$ -th dimension value of the neighboring individual of  $x_{i,j,G}$ , which is defined using a ring topology scheme illustrated in Figure 1. In AMO, Li *et al.* [22] employ the four nearest individuals and the  $i$ -th individual itself for each dimension as its neighborhood and choose one individual in the neighborhood randomly as  $x_{neighborhood,j,G}$ . As an example, Figure 1 shows the neighborhood of  $x_{i,j,G}$  if  $i$ -th individual is  $x_{i,j,G}$ . For the  $j$ -th dimension,  $x_{neighborhood,j,G}$  is selected from the  $(i - 2)$ -th individual, the  $(i - 1)$ -th individual, the  $i$ -th individual, the  $(i + 1)$ -th individual, and the  $(i + 2)$ -th individual.  $\delta$  is a random number produced by a Gaussian distribution with  $N(0, 1)$ .

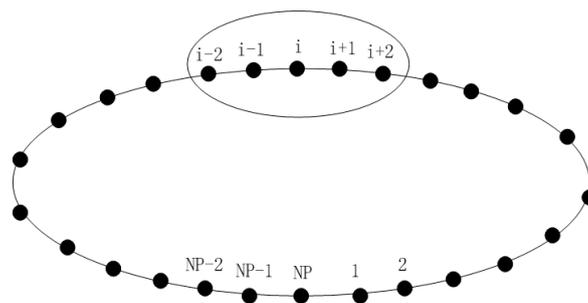


Figure 1. The concept of the local neighborhood of an individual.

The population updating simulates how animals leave the group and new individuals join in the new population, as Equation (6) describes:

$$x_{i,j,G+1} = x_{r1,j,G} + rand_1 \cdot (x_{best,j,G} - x_{i,j,G}) + rand_2 \cdot (x_{r2,j,G} - x_{i,j,G}), \quad (6)$$

where  $x_{r1,j,G}$  is the  $j$ -th dimension value of the individual to be updated, which is chosen randomly in the current population; moreover, different from  $x_{i,j,G}$ ,  $x_{r2,j,G}$  is the  $j$ -th dimension value of another random individual, and  $x_{best,j,G}$  is the  $j$ -th dimension value of the best individual that has been found.  $rand_1$  and  $rand_2$  are two uniformly random numbers between 0 and 1. The algorithm makes the assumption that the number of animals in the population remains unchanged. Therefore, in the updating, it replaces some of the animals with new individual according to a probability  $Pa_i$ , which is related to the fitness of individuals and can be calculated as follows:

$$Pa_i = \frac{sn_i}{NP}, \quad (7)$$

where  $Pa_i$  is the probability value of the  $i$ -th individual,  $NP$  is the number of the individuals in the population, and  $sn_i$  is the sequence number of the fitness of  $i$ -th individual after being sorted by their fitness in descending order, where  $i = 1, 2, \dots, NP$ . According to Equation (7),  $Pa_i$  is 1 if the  $i$ -th individual is of the best fitness, whereas  $Pa_i$  is  $1/NP$  if the  $i$ -th individual has the worst fitness.

---

**Algorithm 1.** Animal Migration Optimization (AMO) algorithm

---

```

1 begin
2 set the generation counter  $G = 0$ ; and randomly initialize  $NP$  individuals denoted as  $X_i$ 
  ( $1 \leq i \leq NP$ ) with  $D_x$  dimensions.
3 evaluate the fitness for each individual.
4 while stopping criteria is not satisfied do
5   for  $i = 1$  to  $NP$  do
6     for  $j = 1$  to  $D_x$  do
7        $x_{i,j,G+1} = x_{i,j,G} + \delta \cdot (x_{neighborhood,j,G} - x_{i,j,G})$ 
8     end for
9   end for
10  for  $i = 1$  to  $NP$  do
11    evaluate the fitness of the offspring  $X_{i,G+1}$ , let  $X_i = X_{i,G+1}$  if  $X_{i,G+1}$  is better than  $X_i$ 
12  end for
13  select  $r1$  and  $r2$  randomly ( $r1 \neq r2 \neq i$ )
14    for  $i = 1$  to  $NP$  do
15      for  $j = 1$  to  $D_x$  do
16        if  $rand > Pa_i$  then
17           $x_{i,j,G+1} = x_{r1,j,G} + rand_1 \cdot (x_{best,j,G} - x_{i,j,G}) + rand_2 \cdot (x_{r2,j,G} - x_{i,j,G})$ 
18        end if
19      end for
20    end for
21    for  $i = 1$  to  $NP$  do
22      evaluate the fitness of the offspring  $X_{i,G+1}$ , let  $X_i = X_{i,G+1}$  if  $X_{i,G+1}$  is better than  $X_i$ 
23    end for
24    memorize the best solution achieved so far
25  end while
26 end

```

---

For each individual and each dimension, a uniformly random number, denoted by *rand*, between 0 and 1 will be produced as the probability to determine whether the individual is reserved or is replaced by a new individual. Therefore, individuals with better fitness will be reserved with higher probability in the next generation, while those with worse fitness will probably be replaced by new individuals. Moreover, the animal with best position will be retained in the next generation. The entire AMO algorithm is described in Algorithm 1 [22].

### 3. The Information Entropy-Based AMO

In this section, we present a modified AMO algorithm. The original AMO algorithm is good at global searching and local searching, and can lead to a satisfactory solution for numeric optimization. However, the data clustering problem is quite different from the benchmarks of numeric optimization problems, as it is easy to see that data in a clustering problem usually has its own distributions. To adapt the AMO algorithm for data clustering, we investigate inherent features and propose an entropy-based AMO algorithm.

#### 3.1. Attribute Information Entropy

It is clear that data in a clustering problem is a collection of points in a multi-dimensional space. In the general case, those points are not randomly positioned, so an attribute in the data may obey a certain distribution. Therefore, it is more reasonable to use different strategies according to the distribution of the attribute rather than to use a single strategy. Information entropy can be used to evaluate the disorder degree of a stochastic variable, so it is a suitable measure of attributes to evaluate their distribution. We will discuss information entropy of attributes in this subsection.

We use the method proposed by Shannon [36] to calculate information entropy, which is usually called Shannon's entropy. Shannon's entropy is used widely in many information measures. Given a clustering data set, we record the number of attributes as  $D$ , and the number of classes (centroids) as  $k$  first. Then, to evaluate Shannon's entropy value, we use  $h_i$  ( $i = 1, 2, \dots, D$ ) to denote entropy of the  $i$ -th attribute, and discretize the attribute values, where each value is approximated to its nearest integer, and then calculate the  $i$ -th attribute entropy as follows:

$$h_i = - \sum_{j=low_i}^{high_i} p_j \log p_j, \quad (8)$$

where  $low_i$  is the minimum integer, and  $high_i$  is the maximum integer after discretization of attribute values.  $p_j$  is the percentage of the  $j$ -th integer of the attribute. Because attribute entropy will be used to control the migration process of the animal population, which will be discussed in the next subsection, we use the maximum possible entropy of the  $i$ -th attribute to normalize Shannon's entropy  $h_i$ . The maximum possible entropy of the  $i$ -th attribute, denoted as  $maxh_i$ , can be calculated as follows:

$$maxh_i = -\log \frac{1}{high_i - low_i + 1}. \quad (9)$$

Finally, the normalized entropy of the  $i$ -th attribute  $normh_i$  can be described as

$$normh_i = \frac{h_i}{maxh_i}, \quad (10)$$

where  $h_i$  and  $maxh_i$  is obtained from Equations (9) and (10), respectively. In the case that  $up_i = low_i$ , we will set  $normh_i$  to 1. Thus, a normalized information entropy vector  $NormH$  can be obtained and  $NormH = (normh_1, normh_2, \dots, normh_D)$ .

### 3.2. The New Animal Migration Method

In this subsection, we present our new migration method. In the original AMO algorithm, an individual may move towards one of its neighbors or move away from the neighbor in the migration operation. This method guarantees diversification of the population. To enhance the convergence speed of the migration step and to improve the effectiveness of global searching, we propose a new method by using an alternative route, where two strategies are employed for migration. One is the original method, where a neighbor is picked up with the method used in the original AMO [22]. The other is newly proposed. It selects  $S$  individuals randomly from the current population ( $1 \leq S \leq NP$ ), and the best one among those  $S$  individuals will be picked up as a reference individual, denoted as  $X_{ref,G}$ . Usually, we set  $S$  to 5. This reference individual as well as the selected neighbor are taken as candidate migration directions, rather than only moving towards their neighbors. Migration direction (moving according to the reference individual or according to the neighbor) is controlled by a randomized approach. To balance the efforts between diversification of the population and convergence speed, information entropy is used here to make decisions. An attribute with large information entropy implies that values of it are uncertain and disordered; thus, searching on this attribute converges slowly, so we accelerate convergence on this attribute by moving it according to the position of the global reference individual with a higher probability than attributes with lower information entropy. Therefore, we present the new migration method as Procedure 1.

In Procedure 1, same as the original method,  $X_{i,G}$  is the current position of the  $i$ -th individual, and  $X_{neighborhood,G}$  is the current position of the selected neighbor. The selection method is the same as the original AMO;  $NP$  is the number of the individuals in the population, and  $D_X$  is the dimension of individuals;  $\delta$  is a random number produced by a Gaussian distribution with  $N(0, 1)$ . Different from the original one, dimensions of an individual are not only processed by its neighbor and may also be processed with  $X_{ref,G}$ , where attribute entropy  $NormH$  controls the direction.  $X_{ref,G}$  will be used if  $rand$  (a uniformly distribution random number between 0 and 1) is smaller than the normalized entropy of the attribute. Otherwise, the individual will move towards or away from  $X_{neighborhood,G}$ .

---

#### Procedure 1. The new animal migration operation

---

```

for  $i = 1$  to  $NP$  do
  Select the best one from  $S$  random individuals as  $X_{ref,G}$ 
  for  $j = 1$  to  $D_X$  do
    if  $rand < normh_j$  then
       $x_{i,j,G+1} = x_{i,j,G} + \delta \cdot (x_{ref,j,G} - x_{i,j,G})$ 
    else
       $x_{i,j,G+1} = x_{i,j,G} + \delta \cdot (x_{neighborhood,j,G} - x_{i,j,G})$ 
    end if
  end for
end for

```

---

### 3.3. The New Population Updating Method

During the population updating of AMO, animals will be replaced by a new individual with a probability approach, and the new individual is produced by Equation (6). To further enhance the convergence speed, we propose a new method to decide the manner of producing new individuals, using an attribute entropy similar to the method of our migration method. Two updating manners are involved in the method: moving towards the best individual and moving towards both the best individual and a random individual. Attributes with a higher entropy will probably move close to those of the best individual. The details of the method are shown as follows:

---

**Procedure 2.** The new population updating operation
 

---

```

select randomly integers  $r1 \neq r2 \neq i$ 
for  $i=1$  to  $NP$  do
  for  $j=1$  to  $D_X$  do
    if  $rand_1 > Pa$  then
      if  $rand_2 < normh_j$  then
         $x_{i,j,G+1} = x_{r1,j,G} + rand_a \cdot (x_{best,j,G} - x_{i,j,G})$ 
      else
         $x_{i,j,G+1} = x_{r1,j,G} + rand_b \cdot (x_{best,j,G} - x_{i,j,G}) + rand_n \cdot (x_{r2,j,G} - x_{i,j,G})$ 
      end if
    end if
  end for
end for
end for

```

---

where  $NP$  is the number of the individuals in the population, and  $D_X$  is the dimension of individuals;  $rand_1$  and  $rand_2$  are both random numbers between 0 and 1 with uniform distribution.  $rand_a$ ,  $rand_b$ , and  $rand_n$  are random numbers between 0 and 1 to control the convergence speed.

### 3.4. The Entire Algorithm for Solving Clustering Problems

With discussions and newly proposed strategies in the above subsections, we present our Entropy-based Animal Migration Optimization (EAMO) combining both the new migration method and the new population updating method.

For solving clustering problems by the EAMO, initializing the population is the first operation. During this process, we set an initial population with  $NP$  animal individuals where an individual is a vector with length  $D_x = D \times K$ , where  $D$  is the number of attributes of the input dataset, and  $K$  is the number of clustering centroids. Positions of  $K$  clustering centroids are encoded into the vector, where the first centroid corresponds to the first  $D$  elements, and the second centroid corresponds to the second  $D$  elements, and so on. Each value in the initial individual vector is produced randomly and uniformly between the maximum value and the minimum value of the corresponding attribute in the input data set. After initialization of population, the EAMO performs optimization iteratively, where Equation (3) is used to evaluate fitness of individuals, until the stopping criterion is satisfied. The detailed description of the algorithm framework is listed as follows.

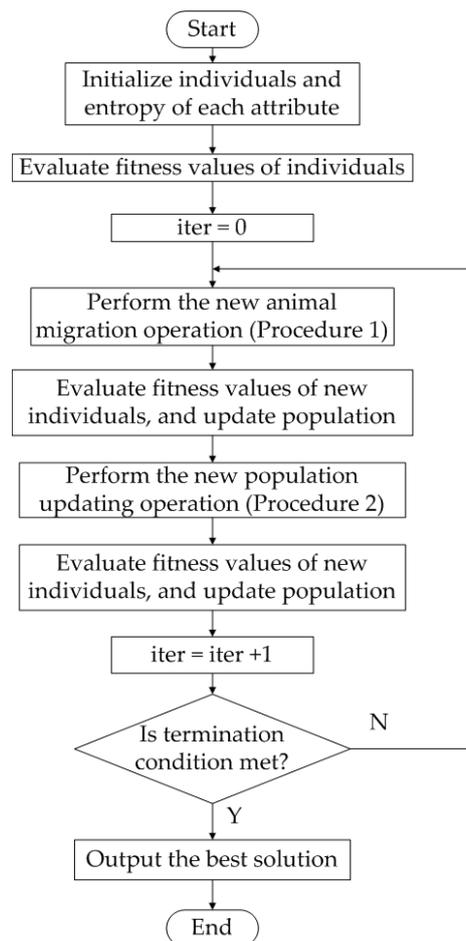
As is shown in Algorithm 2, it starts from generating initial individuals randomly and uniformly in the ranges of attributes from the input data set, and it then calculates the normalized entropy vector  $NormH$ . Afterwards, the algorithm performs optimization iteratively, where the proposed entropy-based migration operation and population updating operation are employed. Attributes will be updated by the population updating operations with probability  $Pa$ , which is the same as the original AMO algorithm. After each migration of an individual, we calculate the fitness of the new location by Equation (3) in Section 1. The new location will replace the current one if the new fitness is better than that of the old location. Identical to the migration operation, new better individuals will replace old ones after each population's updating step. In addition, the best individual found so far will be recorded after all individuals are processed by migration and population updating operations. The algorithm terminates if it achieves the maximum number of iterations, which is the same as the original AMO. The flowchart of the information entropy-based AMO algorithm is shown in Figure 2.

**Algorithm 2.** Information Entropy-based Animal Migration Optimization (EAMO) algorithm

```

1 begin
2 evaluate the information entropy for attributes, and calculate the entropy vector  $NormH$ 
3 set the generation counter  $G = 0$ , and randomly initialize  $NP$  individuals denoted as  $X_i$ 
  ( $1 \leq i \leq NP$ )
4 evaluate the fitness for each individual in the population
5 while stopping criteria is not satisfied do
6   perform the new animal migration operation (Procedure 1)
7   for  $i = 1$  to  $NP$  do
8     evaluate the fitness of the offspring  $X_{i,G+1}$ , let  $X_i = X_{i,G+1}$  if  $X_{i,G+1}$  is better than  $X_i$ 
9   end for
10  perform the new population updating operation (Procedure 2)
11  for  $i = 1$  to  $NP$  do
12    evaluate the fitness of the offspring  $X_{i,G+1}$ , let  $X_i = X_{i,G+1}$  if  $X_{i,G+1}$  is better than  $X_i$ 
13  end for
14  memorize the best solution achieved so far
15 end while
16 end

```



**Figure 2.** The flowchart of the information entropy-based animal migration optimization algorithm.

## 4. Experiments

In this section, we carry out computational experiments to evaluate our algorithm. First, we introduce benchmark data sets we used, and intensive experiments are then performed and compared to other well-known clustering algorithms. We also analyze the effectiveness of our migration methods by comparing with the EAMO algorithm without entropy heuristics.

### 4.1. Data Set

We select 12 well-known benchmark problems from University of California Irvine (UCI) Machine Learning Repository to test those algorithms. Those data sets we choose are frequently used as benchmark for clustering, where the numbers of attributes and the number of classes in each data set are quite different. Table 1 summarizes the main characteristics of those data sets.

Here, we give a brief description of those data sets. All of those data sets come from real-world applications ranging from healthy and medicine to education and criminological investigation. TAE and CMC in Table 1 are the abbreviations of Teaching Assistant Evaluation Data Set and Contraceptive Method Choice Data Set, respectively. The largest data set consists of 1473 objects characterized by 10 attributes, and the data set with the most attributes are Wine Data Set and StatLog (Heart) Data Set, up to 13 attributes. Most data sets have two or three categories, whereas there are six categories in Glass Identification, which is the largest in all data sets. Some of them consist of both categorical attributes and numerical attributes.

**Table 1.** Main characteristics of benchmark data sets.

Name of Data Set	No. of Attributes	No. of Classes	Size of Data Set
Survival	3	2	306
Iris	4	3	150
Scale	4	3	625
TAE	5	3	151
Thyroid	5	3	215
Column	6	3	310
Seeds	7	3	210
Glass	9	6	214
Cancer	9	2	683
CMC	10	3	1473
Wine	13	3	178
Heart	13	2	270

### 4.2. Comparisons with Other Algorithms

In order to demonstrate the effectiveness and performance of the proposed EAMO algorithm, we incorporate our entropy-based method into the source code of AMO and implement the new clustering algorithm within MATLAB (version 7.8, The MathWorks, Inc., Natick, MA, USA). The *k*-means, ABC [18,19], and AMO [22] are considered for comparisons. In addition, we compare our algorithm with recent algorithms such as GSA-KM [37], BH [38], and WK-means [39].

To make a fair comparison, we set population size to 100 for ABC, AMO, and EAMO and each algorithm with maximum 100 iterations as the stop criterion. We perform all experiments on a laptop with an Intel(R) Core(TM) i5-4200M 2.50 GHz CPU, and 4 GB RAM, running Windows 10. Each data set is tested 30 times with random initial solutions. We recorded the result of each run, and counted best and average results of 30 runs to evaluate optimization ability. Standard deviation is also calculated to show the robustness of the algorithms. The results are listed in Table 2.

**Table 2.** Comparison on the results of basic algorithms and Entropy-based Animal Migration Optimization (EAMO).

Data Set	Criteria	<i>k</i> -means	ABC	AMO	EAMO
Survival	Mean (Best)	2903.4982 (2625.1076)	2567.2675 ( <b>2566.9889</b> )	2566.9952 (2566.9890)	<b>2566.9889 (2566.9889)</b>
	Standard Deviation	307.5890	0.4008	$5.9079 \times 10^{-3}$	$4.6754 \times 10^{-7}$
Iris	Mean (Best)	107.6928 (97.3259)	96.7499 (96.6558)	98.1011 (96.6928)	<b>96.6555 (96.6555)</b>
	Standard Deviation	14.8955	0.2213	1.2251	$2.2391 \times 10^{-5}$
Scale	Mean (Best)	1426.1337 (1423.8514)	1425.3914 (1423.8286)	1429.0310 (1426.7837)	<b>1424.5965 (1423.8204)</b>
	Standard Deviation	2.2840	1.5454	1.1745	<b>0.8896</b>
TAE	Mean (Best)	1538.4288 (1505.5616)	1505.7132 (1490.9285)	1496.4112 (1492.5982)	<b>1490.9258 (1490.9258)</b>
	Standard Deviation	47.6131	19.3756	2.9332	$3.0209 \times 10^{-5}$
Thyroid	Mean (Best)	2138.9111 (1987.4110)	1884.8222 (1866.7054)	1930.2179 (1905.8493)	<b>1882.4999 (1866.5277)</b>
	Standard Deviation	166.2917	12.1113	13.9454	<b>11.9139</b>
Column	Mean (Best)	9577.1141 (8990.6570)	8339.0336 (7792.2072)	7929.8714 (7806.5808)	<b>7767.4119 (7767.3986)</b>
	Standard Deviation	1100.9190	354.7481	78.0536	$7.0552 \times 10^{-2}$
Seeds	Mean (Best)	331.6019 (313.2168)	312.0665 (311.8318)	317.3699 (312.2431)	<b>311.7985 (311.7978)</b>
	Standard Deviation	41.0606	0.1724	5.3389	$3.9593 \times 10^{-3}$
Glass	Mean (Best)	266.8226 (215.6775)	241.2736 (221.2115)	289.4854 (277.1934)	<b>227.3104 (214.4359)</b>
	Standard Deviation	35.5765	9.7262	<b>6.7866</b>	11.6376
Cancer	Mean (Best)	2988.0368 (2986.9613)	2965.0793 (2964.6066)	2976.7421 (2964.7393)	<b>2964.3872 (2964.3870)</b>
	Standard Deviation	0.6596	0.6028	21.1566	$6.8615 \times 10^{-4}$
CMC	Mean (Best)	5949.9685 (5703.4379)	5703.3675 (5699.1821)	5748.2252 (5715.3219)	<b>5693.7649 (5693.7253)</b>
	Standard Deviation	557.9685	3.0824	19.0655	<b>0.1186</b>
Wine	Mean (Best)	18,153.4816 (16,555.6794)	16,295.5560 (16,294.1536)	16,301.7626 (16,295.6501)	<b>16,293.3755 (16,292.6728)</b>
	Standard Deviation	1768.3431	1.0243	4.3872	<b>0.8060</b>
Heart	Mean (Best)	11,240.6272 (10,695.7974)	10,623.9668 (10,623.2945)	10,625.2158 (10,623.3220)	<b>10,622.9832 (10,622.9824)</b>
	Standard Deviation	1229.5013	0.7614	1.2657	$2.5003 \times 10^{-3}$

The best values are indicated in bold type.

As can be seen in Table 2, EAMO is able to achieve better average performance than other algorithms ( $k$ -means, ABC, and AMO) for all data sets. For the Survival data set, ABC and EAMO both find the smallest best values, *i.e.*, 2566.9889, but the standard deviation of EAMO is  $4.6754 \times 10^{-7}$ , which is at least 4 orders of magnitude better than the results of other algorithms. Similar results can also be found on some other data sets, such as Iris, TAE, Seeds, Cancer, and Heart. For those data sets mentioned above, EAMO achieves the smallest mean and best values compared to other algorithms, and standard deviations are all several orders of magnitude better than the results of other algorithms. For Glass data set, EAMO achieves better performance, except for the standard deviation. Furthermore, the results in Table 2 also indicate that the EAMO has the strongest robustness with competitive mean values for most clustering problems.

Moreover, we compare our algorithm with clustering algorithms recently proposed. They are GSA-KM [37], BH [38], and WK-means [39]. The results of those three algorithms are from the experiments results in [37–39], respectively. Table 3 shows the comparison results of those algorithms and EAMO.

The results of five data sets in Table 3 have been found from their works [37–39]. Compared with the GSA-KM algorithm, it is obvious that EAMO performs better for four of the data sets (Iris, Cancer, Wine and CMC) out of five compared data sets. Compared with the BH algorithm, EAMO is able to find better results for three data sets (Iris, Cancer and Wine) out of four compared data sets, where the BH algorithm achieves the best solution for the data set of Glass among these four algorithms. In addition, the results of EAMO are also better than those of WK-means for all three compared data sets. Those comparison results prove that EAMO can perform better in most clustering problems than other existing algorithms.

**Table 3.** Comparison on the results of recent clustering algorithms and EAMO.

Data Set	Criteria	GSA-KM [37]	BH [38]	WK-means [39]	EAMO
Iris	Mean	96.689	96.65681	96.6565	<b>96.6555</b>
	Best	96.679	96.65589	<b>96.6555</b>	<b>96.6555</b>
	Worst	96.705	96.66306	96.6704	<b>96.6555</b>
	Standard Deviation	0.0076	0.00173	0.00251	<b><math>2.2391 \times 10^{-6}</math></b>
Cancer	Mean	2965.21	2964.39539	-	<b>2964.3871</b>
	Best	2965.14	2964.38878	-	<b>2964.3870</b>
	Worst	2965.30	2964.45074	-	<b>2964.3902</b>
	Standard Deviation	0.0670	0.00921	-	<b><math>6.8615 \times 10^{-4}</math></b>
Wine	Mean	16,294.31	16,294.31763	16297	<b>16,293.3755</b>
	Best	16,294.25	16,293.41995	16294	<b>16,292.6728</b>
	Worst	16,294.64	16,300.22613	16304	<b>16,295.7335</b>
	Standard Deviation	<b>0.0406</b>	1.65127	2.2019	0.8060
Glass	Mean	214.22	<b>211.49860</b>	-	227.3104
	Best	211.47	<b>210.51549</b>	-	214.4359
	Worst	216.08	<b>213.95689</b>	-	255.7101
	Standard Deviation	<b>1.1371</b>	1.18230	-	11.6376
CMC	Mean	5697.36	-	5751.04	<b>5693.7649</b>
	Best	5697.03	-	5694.6	<b>5693.7253</b>
	Worst	5697.87	-	5988.3	<b>5694.3680</b>
	Standard Deviation	0.2717	-	57.9428	<b>0.1186</b>

The best values are indicated in bold type. The dashed line is filled in the cell if no result can be found.

#### 4.3. Analysis of Entropy-Based Heuristics

In this subsection, we further investigate the contribution of the entropy-based heuristics in our EAMO algorithm. Two AMO-based algorithms are selected for comparison. The first one is a modified EAMO, denoted by EAMO1, by using the newly proposed animal migration operation and the original

population updating operation of the AMO. The second one, denoted by EAMO2, is obtained by using the newly proposed population updating operation and the original animal migration operation of the AMO.

Similar to the experiment in the previous section, 12 data sets are tested with 30 runs for EAMO, EAMO1, and EAMO2. Parameter settings are the same as in the previous experiment. We calculate best and average results as well as the standard deviation to compare with the results of EAMO. Table 4 shows the results of those experiments.

From Table 4, it is clear that EAMO1 and EAMO2 are better than AMO, as they obtain better solutions for most data sets. Between EAMO1 and EAMO2, EAMO1 performs better, as we can see that both mean and best solutions produced by EAMO1 are better than those of EAMO2. On the other hand, EAMO is better than EAMO1 since it performs best on mean results for 10 data sets, whereas EAMO1 performs best only for 3 data sets, and the same mean result is obtained for the data set Survival. Furthermore, EAMO has a good ability to find best results, as it finds the best solutions of 11 data sets, while there are only 6 data sets for which EAMO1 finds the best solutions. Therefore, it can be concluded that both entropy-based heuristic operations make efforts on improving searching effectiveness.

To show statistical results of all data sets, we record the best run by all algorithms for each data set and calculate relative percentage deviation of each run by using Equation (11).

$$RPD = \frac{R - R_b}{R_b}, \quad (11)$$

where  $R$  is the clustering result of a run of an algorithm for a data set,  $R_b$  is the best result of all runs of all algorithms for the data set, and  $RPD$  is the percentage result of  $R$ . By doing so, results of all data sets can be compared together. After that, we show the results made by those algorithms are statistically significant by plotting 95% confidence intervals for the algorithm factor, which is depicted in Figure 3. From it, we can clearly observe that the EAMO has a very good performance overcoming all the remaining methods, such as ABC and AMO.

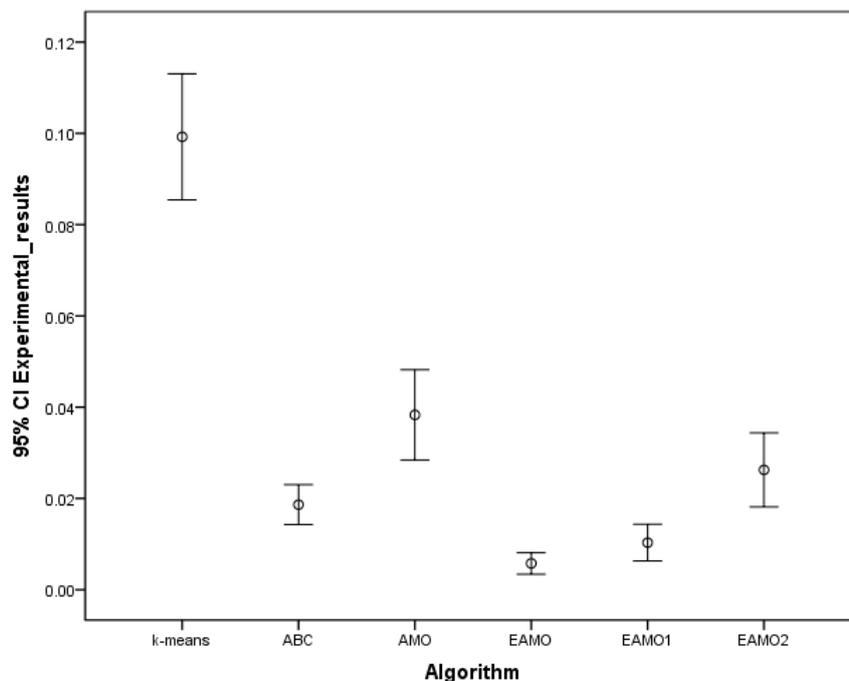


Figure 3. ANOVA tests of the results of all algorithms.

**Table 4.** The results for analyzing entropy-based heuristics in EAMO.

Data Set	Criteria	AMO	EAMO1	EAMO2	EAMO
Survival	Mean (Best)	2566.9952 (2566.9890)	<b>2566.9889 (2566.9889)</b>	2566.9893 ( <b>2566.9889</b> )	<b>2566.9889 (2566.9889)</b>
	Standard Deviation	$5.9079 \times 10^{-3}$	$7.1480 \times 10^{-6}$	$5.8521 \times 10^{-4}$	$4.6754 \times 10^{-7}$
Iris	Mean (Best)	98.1011 (96.6928)	96.6654 ( <b>96.6555</b> )	96.9105 ( <b>96.6555</b> )	<b>96.6555 (96.6555)</b>
	Standard Deviation	1.2251	$5.1261 \times 10^{-2}$	0.3622	$2.2391 \times 10^{-5}$
Scale	Mean (Best)	1429.0310 (1426.7837)	<b>1424.4484 (1423.8204)</b>	1428.4623 (1426.5044)	1424.5965 ( <b>1423.8204</b> )
	Standard Deviation	1.1745	<b>0.8696</b>	1.0406	0.8896
TAE	Mean (Best)	1496.4112 (1492.5982)	1492.3297 ( <b>1490.9258</b> )	1493.2173 (1490.9371)	<b>1490.9258 (1490.9258)</b>
	Standard Deviation	2.9332	6.5013	1.7947	$3.0209 \times 10^{-5}$
Thyroid	Mean (Best)	1930.2179 (1905.8493)	1888.3010 (1868.3156)	1908.2889 (1890.3430)	<b>1882.4999 (1866.5277)</b>
	Standard Deviation	13.9454	<b>9.3286</b>	12.0367	11.9139
Column	Mean (Best)	7929.8714 (7806.5808)	7767.4224 (7767.3987)	7776.3424 (7767.4122)	<b>7767.4119 (7767.3986)</b>
	Standard Deviation	78.0536	$7.7230 \times 10^{-2}$	19.4984	$7.0552 \times 10^{-2}$
Seeds	Mean (Best)	317.3699 (312.2431)	<b>311.7983 (311.7978)</b>	312.3224 (311.7982)	311.7985 ( <b>311.7978</b> )
	Standard Deviation	5.3389	$1.2377 \times 10^{-3}$	1.5837	$3.9593 \times 10^{-3}$
Glass	Mean (Best)	289.4854 (277.1934)	238.1493 (217.1508)	274.2144 (246.1603)	<b>227.3104 (214.4359)</b>
	Standard Deviation	<b>6.7866</b>	18.1387	12.0982	11.6376
Cancer	Mean (Best)	2976.7421 (2964.7393)	2964.4016 (2964.3871)	2964.9262 (2964.3904)	<b>2964.3872 (2964.3870)</b>
	Standard Deviation	21.1566	$4.7387 \times 10^{-2}$	2.0681	$6.8615 \times 10^{-4}$
CMC	Mean (Best)	5748.2252 (5715.3219)	5693.9725 (5693.7275)	5710.9860 (5694.4730)	<b>5693.7649 (5693.7253)</b>
	Standard Deviation	19.0655	0.5741	12.7324	<b>0.1186</b>
Wine	Mean (Best)	16,301.7626 (16,295.6501)	16,294.1431 ( <b>16,292.2713</b> )	16,300.0215 (16,295.7006)	<b>16,293.3755 (16,292.6728)</b>
	Standard Deviation	4.3872	1.5271	2.7806	<b>0.8060</b>
Heart	Mean (Best)	10,625.2158 (10,623.3220)	10,622.9897 (10,622.9825)	10,623.4615 (10,622.9833)	<b>10,622.9832 (10,622.9824)</b>
	Standard Deviation	1.2657	$2.3998 \times 10^{-2}$	0.7456	$2.5003 \times 10^{-3}$

The best values are indicated in bold type.

## 5. Conclusions

In this paper, we present a new AMO algorithm for clustering problems. The information entropy of data in the clustering problems is employed as a heuristic for optimization in the new algorithm. In order to speed up convergence of the proposed algorithm and improve the entire searching performance, we take an alternative manner in the migration step to yield new positions of individuals, where individuals not only move to or away from its neighborhood, but also make movements according to a good individual selected from the entire population. We employ the information entropy of each attribute to determine the probability of moving directions (moving according to the neighbor or the good individual). Furthermore, the population updating method is also modified with similar techniques in migration. Intensive experiments were performed to evaluate effectiveness. The proposed EAMO is tested on 12 well-known benchmark problems from the UCI Machine Learning Repository. Results are analyzed intensively by comparing with both basic algorithms and recently proposed algorithms. The comparison results show that the proposed EAMO algorithm can obtain better solutions than other existing algorithms. In future work, we will consider using information entropy to measure the correlation between attributes to improve clustering algorithms and recent metaheuristic algorithms to solve clustering problems, such as MBO and EHO.

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (No. 61175056, No. 61402070), Natural Science Foundation of Liaoning Province, China (No. 2015020023), and National Students' Innovation and Entrepreneurship Training Project of China (No. 201510151002).

**Author Contributions:** Lei Hou and Jian Gao designed the algorithm and experiments; Lei Hou performed the experiments; Rong Chen analyzed experimental results; Lei Hou and Jian Gao wrote the algorithm section and the experimental section of the paper; Rong Chen organized the structure of the paper. All authors have read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv.* **1999**, *31*, 264–323. [[CrossRef](#)]
2. Xu, R.; Wunsch, D.I.I. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [[CrossRef](#)] [[PubMed](#)]
3. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
4. Aldana-Bobadilla, E.; Kuri-Morales, A. A Clustering Method Based on the Maximum Entropy Principle. *Entropy* **2015**, *17*, 151–180. [[CrossRef](#)]
5. Seret, A.; Verbraken, T.; Baesens, B. A new knowledge-based constrained clustering approach: Theory and application in direct marketing. *Appl. Soft Comput.* **2014**, *24*, 316–327. [[CrossRef](#)]
6. Bayá, A.E.; Larese, M.G.; Granitto, P.M. Clustering using PK-D: A connectivity and density dissimilarity. *Expert Syst. Appl.* **2016**, *51*, 151–160. [[CrossRef](#)]
7. Domenico, M.D.; Insolia, A. Entropic Approach to Multiscale Clustering Analysis. *Entropy* **2012**, *14*, 865–879. [[CrossRef](#)]
8. Filipponea, M.; Camastra, F.; Masulli, F.; Rovetta, S. A survey of kernel and spectral methods for clustering. *Pattern Recognit.* **2008**, *41*, 176–190. [[CrossRef](#)]
9. Barbakh, W.A.; Wu, Y.; Fyfe, C. Review of Clustering Algorithms. In *Non-Standard Parameter Adaptation for Exploratory Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 7–28.
10. Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. NP-hardness of Euclidean sum-of-squares clustering. *Mach. Learn.* **2009**, *75*, 245–249. [[CrossRef](#)]
11. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.
12. Deep, K. A self-organizing migrating genetic algorithm for constrained optimization. *Appl. Math. Comput.* **2008**, *198*, 237–250. [[CrossRef](#)]

13. Homaifar, A.; Qi, C.X.; Lai, S.H. Constrained optimization via genetic algorithms. *Simul. Trans. Soc. Model. Simul. Int.* **1994**, *62*, 242–253. [[CrossRef](#)]
14. Venkatraman, S.; Yen, G.G. A generic framework for constrained optimization using genetic algorithms. *IEEE Trans. Evol. Comput.* **2005**, *9*, 424–435.
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. *IEEE Int. Conf. Neural Netw.* **1995**, *4*, 1942–1948.
16. Hu, X.; Eberhart, R. Solving constrained nonlinear optimization problems with particle swarm optimization. In Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 14–18 July 2002; pp. 203–206.
17. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intel.* **2007**, *20*, 89–99. [[CrossRef](#)]
18. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]
19. Karaboga, D.; Basturk, B. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In *Foundations of Fuzzy Logic and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 789–798.
20. Wang, G.G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2015**, 1–20. [[CrossRef](#)]
21. Wang, G.G.; Deb, S.; Coelho, L.D.S. Elephant Herding Optimization. In Proceedings of the International Symposium on Computational and Business Intelligence, Bali, Indonesia, 7–9 December 2015.
22. Li, X.; Zhang, J.; Yin, M. Animal migration optimization: An optimization algorithm inspired by animal migration behavior. *Neural Comput. Appl.* **2014**, *24*, 1867–1877. [[CrossRef](#)]
23. Li, X.; Zhang, X.; Yin, M.; Wang, J. A genetic algorithm for the distributed assembly permutation flowshop scheduling problem. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 3096–3101.
24. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [[CrossRef](#)]
25. Guedria, N.B. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **2016**, *40*, 455–467. [[CrossRef](#)]
26. Li, X.; Yin, M. Self-adaptive constrained artificial bee colony for constrained numerical optimization. *Neural Comput. Appl.* **2014**, *24*, 723–734. [[CrossRef](#)]
27. Kao, Y.T.; Zahara, E.; Kao, I.W. A hybridized approach to data clustering. *Expert Syst. Appl.* **2008**, *34*, 1754–1762. [[CrossRef](#)]
28. Cura, T. A particle swarm optimization approach to clustering. *Expert Syst. Appl.* **2012**, *39*, 1582–1588. [[CrossRef](#)]
29. Krishnasamy, G.; Kulkarni, A.J.; Paramesran, R. A hybrid approach for data clustering based on modified cohort intelligence and K-means. *Expert Syst. Appl.* **2014**, *41*, 6009–6016. [[CrossRef](#)]
30. Karaboga, D.; Ozturk, C. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Appl. Soft Comput.* **2011**, *11*, 652–657. [[CrossRef](#)]
31. Zhang, S.; Zhou, Y. Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis. *Discret. Dyn. Nat. Soc.* **2015**. [[CrossRef](#)]
32. Zou, W.; Zhu, Y.; Chen, H.; Sui, X. A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm. *Discret. Dyn. Nat. Soc.* **2010**, *2010*, 1038–1045. [[CrossRef](#)]
33. Jing, L.; Ng, M.K.; Huang, J.Z. An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1026–1041. [[CrossRef](#)]
34. Liang, J.; Zhao, X.; Li, D.; Cao, F.; Dang, C. Determining the number of clusters using information entropy for mixed data. *Pattern Recognit.* **2012**, *45*, 2251–2265. [[CrossRef](#)]
35. Cheung, Y.M.; Jia, H. Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognit.* **2013**, *46*, 2228–2238. [[CrossRef](#)]
36. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
37. Hatamlou, A.; Abdullah, S.; Nezamabadi-Pour, H. A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm Evol. Comput.* **2012**, *6*, 47–52. [[CrossRef](#)]

38. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [[CrossRef](#)]
39. Boobord, F.; Othman, Z.; Bakar, A.A. A WK-Means Approach for Clustering. *Int. Arab J. Inf. Technol.* **2015**, *12*, 489–493.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).