Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce ☆

Ashish Kumar Tripathi, Kapil Sharma *, Manju Bala

ARTICLE INFO

Article history:
Received 30 January 2018
Received in revised form 9 April 2018
Accepted 6 May 2018
Available online xxxx

Keywords:
Clustering method
Grey-wolf optimizer
Hadoop
MapReduce

ABSTRACT

With advancement of the technology, data size is increasing rapidly. For making intelligent decisions based on data, efficacious analytic methods are required. Data clustering, a prominent analytic method of data mining, is being efficiently employed in data analytics. To analyze massive data sets, the improvement in the traditional methods is the urge of todays scenario. In this paper, an efficient clustering method, MapReduce based enhanced grey wolf optimizer (MR-EGWO), is presented for clustering large-scale data sets. The proposed method introduced a novel variant of grey wolf optimizer, Enhanced grey wolf optimizer (EGWO), where the hunting strategy of grey wolf is hybridized with binomial crossover and lévy flight steps are inducted to enhance the searching capability for pray. Further, the proposed variant is used for optimizing the clustering process. The clustering efficiency of the EGWO is tested on seven UCI benchmark datasets and compared with the five existing clustering techniques namely K-Means, particle swarm optimization (PSO), gravitational search algorithm (GSA), bat algorithm (BA) and grey wolf optimizer (GWO). The convergence behavior and consistency of the EGWO has been validated through the convergence graph and boxplots. Further, the proposed EGWO is parallelized on the MapReduce model in the Hadoop framework and named MR-EGWO to handle the large-scale datasets. Moreover, the clustering quality of the MR-EGWO is also validated in terms of F-measure and compared with four MapReduce based state-of-the-art namely; parallel K-Means, parallel K-PSO, MapReduce based artificial bee colony optimization (MR-ABC), dynamic frequency based parallel k-bat algorithm (DFBPKBA). Experimental results affirm that the proposed technique is promising and powerful alternative for the efficient and large-scale data clustering.

1. Introduction

Clustering is the prominent approach of unsupervised learning and considered as part and parcel of data engineering applications, such as image segmentation, data mining, information retrieval system, anomaly detection, medicine, computer vision and construction management [1–4]. Over the past years, several algorithms for data clustering have been introduced in the literature to handle the diversity in data and different sets of application requirements. K-means, one of the simplest and popular algorithm, has been employed for unfolding the various clustering problems [5], [6]. However, the results of K-means algorithm are highly dependent on initial cluster centroids and it's probability of trapping into local optima is high [7].

To mitigate this issue, various metaheuristic-based clustering methods have been proposed in literature. Maulik et al. [8] used the ability of genetic algorithm to find the best centroid in the feature space so that the compactness of the resulting clusters is optimized. Sharma et al. [11] proposed bat algorithm based method for optimizing the clustering process. The proposed method was also parallelized using MapReduce to handle large data sets. Hatamlou et al. [13] introduced gravitational search algorithm based clustering method which used K-means algorithm for initializing the center heads. Karaboga et al. [9] proposed a novel artificial bee colony based method for clustering the multivariate data. Kumar et al. [12] introduced a clustering algorithm which imitate the hunting behavior of grey wolves. Cura et al. [10] developed particle swarm optimization based clustering method and solved the web application problems. In 2017, Ebrahimi et al. [14] introduced an adaptive meta-heuristic search based method to optimal cluster the sensors, deployed in the environment of Internet of Things. Pal et al. [16] proposed enhanced bio-geography based data clustering algorithm. The proposed algorithm has witnessed better performance as compared to the traditional clustering algorithms. Further, an exponential k-best gravitation search algorithm was introduced by Mittal and Saraswat to find the optimum threshold to perform multilevel image segmentation [15]. Pandey et al. [17]

☆ This article belongs to Special Issue: HEST4BDAA.

* Corresponding author.
E-mail address: kapil@ieee.org (K. Sharma).

https://doi.org/10.1016/j.bdr.2018.05.002
2214-5796/

proposed hybrid cuckoo search method for clustering twitter data for the sentiment analysis of the users. However, the above clustering algorithms fail to perform efficiently on large datasets in terms of memory space and the time complexities due to their sequential execution. For alleviating computation performance on large dataset, parallel and distributed computation has exhibited attractive solutions. With the years of progress in parallelization tools, apache hadoop is a widely used parallelization tool.

Hadoop [18] is an open source platform, developed and managed by Apache for handling large datasets using distributed processing. Hadoop works with its own file system referred as HDFS (hadoop distributed file system) and, is capable of processing zeta bytes of data with commodity hardware [19]. MapReduce [20] provides the parallel computation platform and has successfully leveraged the strengths of meta-heuristics algorithms for the analyses of large-scale datasets [21,23,24]. Gong et al. [25] studied the different distributed evolutionary models and appreciated the simplicity of MapReduce architecture for solving various computation intensive problems. Thus, researchers have worked upon MapReduce based parallel meta-heuristic algorithms in the last five years. A hybrid K-PSO method with MapReduce architecture was proposed by Wang et al. [23] to cluster massive datasets. Banharnsakun [31] proposed MapReduce based artificial bee colony algorithm (MR-ABC) for clustering large datasets. Tripathi et al. [33] proposed dynamic frequency base K-Bat algorithm for handling massive datasets (DFBPKBA). In the proposed method, the frequency of the bats was dynamically changed to improve the clustering accuracy and MapReduce architecture was used to handle large datasets. Zhao et al. [34] were successful in mining knowledge from the big data through the parallel version of K-Means algorithm. Khezr et al. [21] studied various distributed models of the nature inspired algorithms and concluded that the Hadoop MapReduce model is one of the widely used platform for the parallel processing of large datasets due to simplicity and robustness. As "No free lunch" theorem [22] obviates the claim of Idle meta-heuristic algorithm for all set of optimization problems, this paper utilizes the merits of GWO to cluster massive datasets in parallel.

The grey wolf optimizer (GWO), a meta-heuristic algorithm [40], is inspired by the hunting behavior of Grey wolves. It has outperformed existing meta-heuristic algorithms namely; particle swarm optimization (PSO), evolution strategy (ES), gravitational search algorithm (GSA), differential evolution (DE) on standard benchmark problems [40]. The GWO algorithm has been widely used in a number of applications in last three years. Emary et al. [45] used the binary version of GWO to perform optimal feature selection. Komaki and Kayvanfar [49] optimized the flow shop scheduling tasks by applying GWO. Song et al. [55] unfolded GWO for the optimal tuning of the surface waves parameters. Moreover, GWO has also been applied for solving the power dispatch and mixed heat task for power systems [48]. Medjahed et al. [47] introduced a GWO-based method for selection of hyper-spectral bands. Fergany and Hasanien [54], demonstrated the efficiency of GWO on optimal power flow (OPF) problem. On the same footnote, GWO was drafted to design the wide area power system stabilizer (WAPSS) [56]. To solve the economic dispatch problem, Jayabarathi et al. [46] introduced mutation and crossover mechanisms in GWO. Guha [50] employed GWO in power systems for optimizing the load frequency control (LFC). Song et al. utilized the strengths of GWO to find the optimal parameters of the surface waves. [52]. GWO optimizer is also used effectively for training the multi-layer perceptrons [53]. Shima Amirsadri et al. [51] proposed a new variant of GWO using lévy flights in combination with back propagation for training the neural network.

Despite of wide applicability, GWO has limitation of lack of population diversity. This results in slow convergence rate and risk of trapping into local optima [41]. To improves its search precision, a novel variant of GWO, Enhanced grey wolf optimizer (EGWO), is proposed in this paper by incorporating the following capabilities.

- Lévy Flight steps: To magnify search for prey.
- Binomial crossover: To inflate the attack to pray.

The overall contribution of this paper has been divided into three folds. First, a novel clustering method is proposed based on the new variant of GWO. Second, the efficiency of proposed variant is studied on clustering problem. Third, the proposed method is parallelized using MapReduce architecture and named MR-EGWO for efficacious clustering of large datasets. The empirical analysis of EGWO has been done on seven UCI datasets and validated against five clustering algorithms namely K-Means [5], PSO [10], GSA [13], BA [11] and GWO [12] in terms of Mean and Best of intra cluster distance. The convergence behavior of EGWO is discussed along with the box plots to visualize its consistency. Moreover, the clustering performance of MR-EGWO is also validated in terms of F-measure by comparing with four MapReduce based parallel state-of-the-art namely: PK-Means, parallel K-PSO, MR-ABC, DFBPKBA. To demonstrate the parallel computation performance, the proposed method (MR-EGWO) has been analyzed on four large scale datasets and asserted through speedup graphs.

The remainder of this paper is organized as follows. Section 2 briefs the basics of clustering and GWO algorithm. In section 3, the clustering process using the proposed algorithm along with its parallelization using MapReduce is explained. Section 4 explains the experimental environment settings and the simulation results. Finally, the paper in concluded in section 5.

2. Background

2.1. Data clustering approach

Clustering of a dataset in t-dimensional space is the process of assembling of N data objects into K groups on the basis of resemblance [31]. Clustering partitions the data objects iteratively into K groups (clusters) in such a way, that the data objects within the same group have maximum resemblance. Further, data clustering is a type of unsupervised learning approach, that means data objects are grouped on the basis of structure of the data, without any training. Whereas, in supervised learning like classifications, data objects are classified based on the training set using labeled data. The proposed EGWO performs clustering with the known number of clusters. The summation of intra-cluster distance of K clusters is chosen as the criterion for the evaluation of the quality of the clustering. Let $Z = (z_1, z_2, z_3, \cdots, z_n)$ is a collection of N data objects where all the data object are represented in t dimensional space. The data objects are represented by a matrix of $Z_{n\times t}$ having n rows and t columns where each row vector describes one data object. The clustering process allocates the set of N data objects to K clusters and find a set of cluster centroids, $C = \{C_1, C_2, \cdots, C_k\}$ with the aim of minimizing the sum of squired euclidean distance between each data object Z_i and its centroids C_i to which it belongs. Generally, clustering process satisfies the following properties:

- Each and every cluster must have at minimum one data object, i.e., $C_i \neq \phi, \forall i \in \{1, 2, 3, \cdots, k\}$.
- Each data object certainly be part of a cluster.
- No data object can be part of more than one cluster, i.e., $C_q \cap C_r = \phi, \forall q \neq r$ and $q, r \in \{1, 2, 3, \cdots, k\}$.

A dataset is grouped based on the above three conditions and the quality of clustering is evaluated in terms of the fitness value. The sum of squared Euclidean distance [38] is one of the famous func-

Please cite this article in press as: A.K. Tripathi et al., A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce, Big Data Res. (2018), https://doi.org/10.1016/j.bdr.2018.05.002

tion used for the evaluating the quality of the clustering, which is calculated using Eq. (1).

$$f(Z,C)=\sum_{l=1}^{k}\sum_{Z_i\epsilon C_l} d(Z_i,C_l)^2 \tag{1}$$

where $d(Z_i, C_l)$ is the measure of diversity between the data object Z_i and centroid of the cluster C_l. For calculating the dissimilarity between data objects, many distance metrics have been proposed. Euclidean distance is one of the popular distance metric available to compute the dissimilarity between the data objects. Given two data objects Z_i and Z_j with t dimensions, the euclidean distance $d(Z_i, Z_j)$ is calculated as by equation Eq. (2).

$$d(Z_i,Z_j)=\sqrt{\sum_{t=1}^{t}(z_i^t-z_j^t)^2} \tag{2}$$

2.2. Grey wolf optimizer

Grey wolf optimizer (GWO) is a meta-heuristic algorithm proposed by Mirjalili et al. [40] which imitate the hunting mechanism of the grey wolves. In GWO, grey wolves are grouped into *alpha* (α), *beta* (β), *delta* (δ) and *omega* (ω) according to their social hierarchy. The best three grey wolves are considered as *alpha*, *beta*, *delta* and renaming grey wolves are termed as *omega*. The *alpha* wolves are the commanding one and all other grey wolves follows their instructions. The second category of the wolves belonging to the *Beta* category are responsible for helping *alpha* in their decision making. *Omega* are the lowest ranked grey wolves.

In the grey wolf algorithm, hunting is escorted by *alpha*, *beta* and *delta* while *omega* wolves are responsible for encircling the prey to find better solution. The encircling operation performed by the grey wolves is mathematically defined by Eq. (3) and (4):

$$\overrightarrow{D}=|\overrightarrow{C}.\overrightarrow{X}_p(i)-\overrightarrow{X}(i)| \tag{3}$$
$$\overrightarrow{X}(i+1)=\overrightarrow{X}_p(i)+\overrightarrow{A}.\overrightarrow{D} \tag{4}$$

where X_p is the location of the prey, $X(i)$ is the location of the grey wolf at ith iteration. $\overrightarrow{A}$, and $\overrightarrow{C}$ are coefficient vectors and computed using Eqs. (5) and (6) respectively.

$$\overrightarrow{A}=2\overrightarrow{a}.\overrightarrow{r_1}-a. \tag{5}$$
$$\overrightarrow{C}=2.\overrightarrow{r_2}. \tag{6}$$

where $\overrightarrow{a}$ is coefficient vector which is reduced linearly from 2 to 0 with the increasing number of iterations and r_1, r_2 are the random numbers between $[0, 1]$.

$$\overrightarrow{D}_\alpha=|\overrightarrow{C}_1.\overrightarrow{X}_\alpha-\overrightarrow{X}| \tag{7}$$
$$\overrightarrow{D}_\beta=|\overrightarrow{C}_2.\overrightarrow{X}_\beta-\overrightarrow{X}| \tag{8}$$
$$\overrightarrow{D}_\delta=|\overrightarrow{C}_3.\overrightarrow{X}_\delta-\overrightarrow{X}| \tag{9}$$

Further, Eqs. (7), (8) and (9) define the estimated span around the current position and *alpha*, *beta* and *delta*, respectively. After estimating of distances, the final position of the ω wolves is determined by Eq. (10). Where, $\overrightarrow{A}_1$, $\overrightarrow{A}_2$, $\overrightarrow{A}_3$ represents the random vectors, i shows the current iteration number and the vectors $\overrightarrow{X}_1$, $\overrightarrow{X}_2$, $\overrightarrow{X}_3$, are defined by Eqs. (11), (12), and (13) respectively.

$$\overrightarrow{X}(i+1)=\left[\frac{\overrightarrow{X}_1+\overrightarrow{X}_2+\overrightarrow{X}_3}{3}\right] \tag{10}$$
$$\overrightarrow{X}_1=\overrightarrow{X}_\alpha-A_1.D_\alpha \tag{11}$$
$$\overrightarrow{X}_2=\overrightarrow{X}_\beta-\overrightarrow{A}_2.\overrightarrow{D}_\beta \tag{12}$$
$$\overrightarrow{X}_3=\overrightarrow{X}_\delta-\overrightarrow{A}_3.\overrightarrow{D}_\delta \tag{13}$$

3. Proposed method

To deal the problems of large dataset clustering, a novel method, Map-reduce based enhanced grey wolf optimizer (MR-EGWO), is proposed. MR-EGWO leverages the strengths of a novel variant of grey wolf optimizer, enhanced grey wolf optimizer (EGWO), for efficient data clustering. In this section, first a detailed description of the enhanced grey wolf optimizer (EGWO) is presented followed by its parallel version, Map-reduce enhanced grey wolf optimizer (MR-EGWO) is discussed.

3.1. Enhanced grey wolf optimizer (EGWO)

The success of a meta-heuristic algorithm depends upon the equilibrium between exploration and exploitation [32]. The GWO algorithm has limitations of slow convergence rate and risk of trapping into local optima due to the lack of diversity in the wolves for certain cases [41]. These limitations can be overcome by the increase of diversification and intensification of the search space. Therefore, in this paper a novel variant of GWO named enhanced grey wolf optimizer (EGWO) is proposed. The proposed method is empowered with the advantages of lévy [42] flights and binomial crossover [32] to improve the exploration and exploitation capabilities. The EGWO introduces two new phases to relieve the above mentioned problem.

3.1.1. Inflated attack to pray using binomial crossover

As the intensification of the population around the current best solution inflates the generation of optimal solutions. The exploitation in the proposed variant is enhanced by including one of the popular and widely used binomial crossover operator present in the literature. As *alpha* wolf defines the current best position, its position can be used to define the better position of other wolfs. Hence, binomial crossover operator is performed between the *alpha* and the $X(i)$ to inflate the attack to the pray. The updated position (UP) of grey wolves is defined in Eq. (14).

$$UP_i^j=\begin{cases}\overrightarrow{X}_\alpha^j & (K\le C)\\ \overrightarrow{X}_i^j & (K>C)\end{cases} \tag{14}$$

Where UP_i^j is the position of the ith grey wolf in jth dimension, K is the random number between $[0, 1]$. $C \in [0, 1]$ is crossover constant.

3.1.2. Magnified search for pray using on lévy flight

In GWO, the problem of stagnation still prevails in some cases, since the position updation of a wolf is determined solely by the positions of leader wolves namely, alpha, beta, and delta. Correspondingly, GWO results in immature convergence. To enhance the exploration capability, the proposed EGWO uses the concept of lévy flight to update the position of each wolf. As lévy flight defines steps of random lengths drawn from the lévy distribution [39], the chances of exploring the search space increases. This paper uses the Mantegna algorithm [42] to generate steps of random length. The Eq. (15) depicts the formulation of step length z defined by Mantega's algorithm.

$$z=\left[\frac{r}{|s|^{1/\beta}}\right] \tag{15}$$

where, $\beta \in (0, 2]$ is lévy index and r and s are variables following normal distribution of $N(0,\sigma_r^2)$ and $N(0,\sigma_s^2)$, respectively. The σ_r is calculated by Eq. (16) while σ_s is always 1.

$$\sigma_r=\left[\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma[(1+\beta)/2]2^{(\beta-1)/2}}\right]^{1/\beta} \tag{16}$$

Please cite this article in press as: A.K. Tripathi et al., A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce, Big Data Res. (2018), https://doi.org/10.1016/j.bdr.2018.05.002

where, $\Gamma()$ is called Gamma function and defined by Eq. (17).

$$\Gamma(1+\beta) = \int_0^\infty t^\beta e^{-t} dt \tag{17}$$

In the proposed phase, each grey wolf takes lévy flight for the search of the prey and updates its position using Eq. (18).

$$\overrightarrow{X}_{t+1} = \overrightarrow{X}_t + estep_t \tag{18}$$

where, $\overrightarrow{X}_t$ is the position of the grey wolf at tth iteration, $\overrightarrow{X}_\alpha$ represents the position of *alpha* wolf and $estep_t$ at a particular iteration t defines the lévy flight step size and calculated by Eq. (19).

$$estep_t = 0.01 \times z \times (\overrightarrow{X}_t - \overrightarrow{X}_\alpha) \tag{19}$$

3.2. EGWO based clustering

Furthermore, the proposed enhanced grey wolf optimizer (EGWO) is elucidated for the clustering problem. In EGWO based clustering, the position X of each grey wolf represents a set of cluster centroids $(C_1, C_2, C_3, \cdots, C_K)$ for K clusters. The minimization of intra-cluster distance is considered as the cost function and formulated in Eq. (20).

$$f(Z, C) = \sum_{l=1}^{k} \sum_{Z_i \epsilon C_l} d(Z_i, C_l)^2 \tag{20}$$

The optimal clusters corresponds to the position of the *Alpha* wolf. The pseudo-code of the EGWO based clustering method is described in Algorithm 1.

The computation time of the EGWO based clustering is proportional to the size and the number of clusters in the dataset. In this paper, EGWO generates the optimal cluster centroids with $O(N \times K \times t)$ operations for t iterations, where N is the number of data objects and K corresponds to the required number of clusters. Therefore, for P population size, the total time complexity of the proposed clustering method is $O(P \times N \times K \times t)$.

3.3. Parallelization of the EGWO using MapReduce architecture

To demonstrate the applicability of EGWO on large dataset, a parallel version of EGWO algorithm using Hadoop MapReduce framework, MapReduce based EGWO (MR-EGWO), is presented. MR-EGWO works in two phases; EGWO-Map and EGWO-Reduce. Initially, MapReduce framework divides the large datasets into smaller chunks and distribute them uniformly among the hadoop nodes. Further, each data sample is converted into key/value pairs by the record reader. The MR-EGWO map phase, then, processes the input key/value pairs with cluster centroids in parallel and finds the centroid index of each data object. The pseudo-code of the MR-EGWO Map phase is presented in Algorithm 2. The output of this phase is the another set of key/value pair where key consists of $\{gwoID, centroidID\}$, while the distance of data object with the respective centroid-ID defines the value component. Further, the reduce function of the MR-EGWO reduce phase merges all the computed values with identical key's and computes the corresponding fitness value for each grey-wolf. Algorithm 3 presents the pseudo-code of the EGWO-reduce function. The *alpha*, *beta*, and *delta* wolves are updated along with the position of each grey wolf

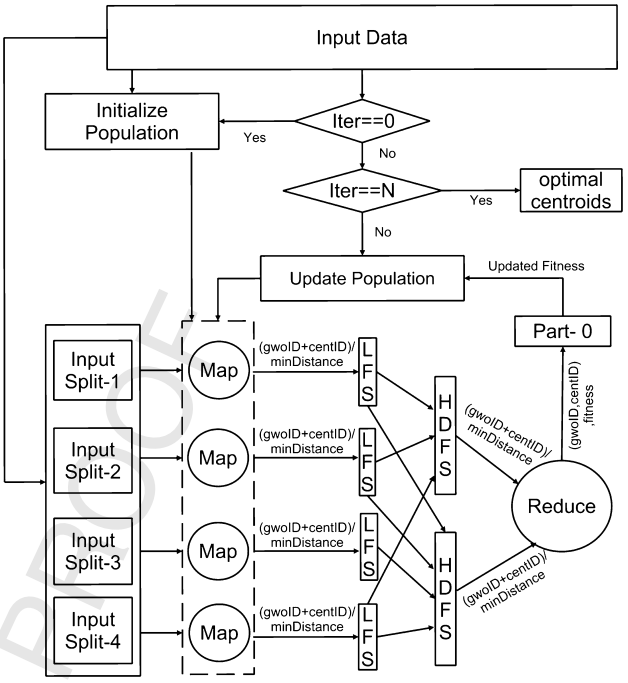

Fig. 1. MapReduce architecture MR-EGWO for data clustering.

Algorithm 1: Enhanced grey wolf optimizer based clustering.

Input: Data file having Z data objects with t dimensions and K Number of clusters.
Ouput: Final centroids position. /* The location of α after termination of algorithm represents centroids position*/

1: Generate initial population of N grey wolves.
2: Initialize parameters a, i, A, C, maximum number of iteration $MaxItr$.
3: Evaluate the fitness of each grey wolf using Eq. (1).
4: Set top three grey wolves according to the fitness as $\overrightarrow{X}_\alpha$, $\overrightarrow{X}_\beta$ and $\overrightarrow{X}_\delta$.
5: **while** ($MaxItr$) *or* (*centroid movement becomes zero*) **do**
6: **for** each grey wolf **do**
7: Update the position of each grey wolf defined by Eq. (10)
8: Perform binomial cross over determined by Eq. (14).
9: Determine the new position of each grey wolf using lévy flight defined by Eq. (18).
10: Upgrade the values of a, A, C.
11: Calculate the fitness of each grey wolf.
12: Update $\overrightarrow{X}_\alpha$, $\overrightarrow{X}_\beta$, and $\overrightarrow{X}_\delta$.
13: **end for**
14: $i = i + 1$;
15: **end while**
16: Return $\overrightarrow{X}\alpha$ /*the position of alpha is the final centroid position*/

Algorithm 2: MR-EGWO map.

```
Map (Key: recordId, Value: Record)
Initialization:
key=record-ID
value=record
read(gwoPopulation);
for each wolf in gwo-population;
gwoID =retrieve-gwoID(gwoPopulation)
centroidArray =retrieve-centroids(gwoPopulation) /* wolf position represents
centroids */
minDistance= getMinD(record, centroidArray);
/* The getMinD() function to get minimum distance is described below */
for each centroid do
    distance = get distance of ith centroid from record
    if (distance < minDistance) then
        minDistance=distance
        centroid-ID = i / i represents index of the centroid array having
        minimum distance
    end if
end for
updated-key= gwoID+centroidID;
end for
write (updated-key, minDistance);
```

Algorithm 3: MR-EGWO reduce.

```
Reduce (Key:(gwoID, centroidID), value-list: minDistance)
Initialization
fitness=0;
for each value in minDistance list do
        minDistance=retrieve-minDistance(value-list)
        fitness=+minDistance
        end for
        write(key, fitness)
end for
```

Table 1
Dataset description.

Dataset	NOC	NOF	NOI
Iris	3	4	150
Wine	3	13	178
Seeds	3	7	210
Glass	6	9	214
Cancer	2	9	638
Balance	3	4	625
Haberman	2	3	306

NOC: Number of clusters.
NOF: Number of features.
NOI: Number of instances.

Table 2
Parameter values of the proposed and considered algorithms.

Parameter name	K-means	PSO	GSA	BAT	GWO	EGWO
Population size (pop)	–	40	40	40	40	40
Number of iterations (itr)	500	500	500	500	500	500
Inertial constant (w)	–	0.5	–	–	–	–
Cognitive constant (c1)	–	1	–	–	–	–
Social constant (c2)	–	1	–	–	–	–
Gconstant (G0)	–	–	20	–	–	–
Alpha (α)	20	–	.9	2	2	–
fmin	–	–	–	0	–	–
fmax	–	–	–	2	–	–
gamma (γ)	–	–	–	.9	–	–
r0	–	–	–	.9	–	–
Crossover constant (C)	–	–	–	–	–	2

according to the EGWO 1. This marks one iteration of the MR-EGWO and this process is continued until the stopping criterion is reached. The complete architecture of the MR-EGWO for data clustering is shown in Fig. 1.

4. Experimental results

The proposed work is evaluated in two folds. First, EGWO is validated for clustering in terms of intra-cluster distance and convergence behavior. The comparison is made with k-means and four meta-heuristic algorithms for clustering namely; GSA, PSO, BA, and GWO. Second, the effectiveness of the MapReduce based MR-EGWO is vindicated in terms of F-measure against the four state-of-the-art MapReduce based clustering methods namely parallel K-means (PKmeans) [34], parallel K-PSO based on MapReduce (parallel K-PSO) [23], MapReduce based artificial bee colony optimization for large scale data clustering (MR-ABC) [31] and Dynamic frequency based parallel K-Bat algorithm (DFBPKBA) [33]. The speedup behavior of MR-EGWO is also studied by incrementing number of nodes in each run.

4.1. Performance analysis of EGWO based clustering

The proposed EGWO algorithm is tested on seven benchmark datasets taken from UCI repository [43] and results are compared with K-means, PSO, GSA, BA and GWO. Table 1 summarizes the seven considered benchmark datasets. The simulation is carried out for 30 runs on a system with Matlab 2015a, intel core i3 processor, 2.80 GHz frequency, 4 GB of RAM and 500 GB hard-disk. Table 2 details the parameter setting of the experimentation.

Table 3 defines the best and mean fitness values attained by the proposed and considered methods over 30 runs. It can be observed from the Table 3 that, EGWO outperformed all five methods on all the datasets in terms of best fitness value. For mean fitness value, EGWO has surpassed results for wine, seeds, glass and cancer. However, GWO has competitive results on Iris and Balance datasets while PSO performed well on Haberman dataset.

Moreover, to validate the performance difference in the proposed and tested methods, a non parametric statistical test, Wilcoxon rank sum test, is conducted at 5% level of significance. Table 4 contains the p-value and SGFT (*significance*) of each method. The null hypothesis is rejected if p-value < 0.05 and symbolized by '+' or '−', else, it is accepted and represented by '=' symbol. The '+' indicates that the method is different and significantly good while '−' shows that it is different and significantly poor. It can be observed from Table 4 that p-value < 0.5 on all the datasets. Correspondingly, it is assured that the EGWO is significantly different from the considered methods except GSA for balance dataset.

To demonstrate the improvement in exploration and exploitation trade-off, convergence behavior of the EGWO and considered methods are illustrated on two datasets, namely wine and seeds as shown in Fig. 2. Horizontal axis represents the iteration numbers while corresponding fitness values are aligned along the vertical axis. It can be visualized from Fig. 2 that EGWO prefers exploration at early stage of iterations and then lessen its exploration rate to perform the exploitation. In the later stage, this decline exploits the search space well for finding the optimal solution. Hence, it is pertinent from the convergence graphs that EGWO improves the exploration and exploitation abilities contrary to GWO. Further, box-plots in Fig. 3 represent the consistency of the clustering

Table 3
Best and mean fitness value over 30 runs.

Dataset	Criteria	K-means	PSO	GSA	BA	GWO	EGWO
Iris	Best	97.34084	96.78998	96.65548	96.65552	96.65826	**96.65548**
	Mean	106.33437	97.13691	96.67516	99.53097	**99.12574**	99.55645
Seeds	Best	587.31957	312.68370	311.79804	311.79816	311.88200	**311.79804**
	Mean	588.10457	313.85971	311.79804	315.41951	312.09220	**311.79804**
Glass	Best	292.75724	238.51144	286.11855	243.70331	265.81420	**214.44399**
	Mean	325.54765	257.06514	316.71044	264.10417	302.04114	**242.68894**
Cancer	Best	19323.17382	2969.23958	2970.17834	2964.38718	2964.390179	**2964.38697**
	Mean	19323.17693	2976.15128	2994.77937	3032.42259	2964.39495	**2964.38697**
Balance	Best	3472.32142	1423.96787	1423.82042	1424.04307	1423.82106	**1423.82040**
	Mean	3493.80000	1424.62818	1424.51503	1426.28547	**1423.82963**	1424.20479
Haberman	Best	30507.02076	2566.99548	2566.98989	2566.98889	2567.02562	**2566.98889**
	Mean	32271.96242	**2567.12294**	2582.08625	2648.88585	2590.77309	2637.34900
Wine	Best	2370689.68700	16298.98906	17038.59226	16371.05448	16307.09242	**16292.18465**
	Mean	2484626.08700	16305.11720	17709.43544	16865.72325	16318.41351	**16292.35069**

Table 4
Results of Wilcoxon test for statistically significance level at $\alpha = 0.05$.

Dataset	EGWO-K-means		EGWO-PSO		EHGWO-GSA		EGWO-BAT		EGWO-GWO	
	P-value	SGFT	P-value	SGFT	P-value	SGFT	P-value	SGFT	P-value	SGFT
Iris	4.45E−08	+	6.76E−05	+	2.40E−09	+	4.11E−06	+	1.42E−05	+
Seeds	2.78E−09	+	3.11E−11	+	2.55E−11	+	3.01E−09	+	3.11E−10	+
Glass	4.41E−08	+	6.28E−06	+	3.02E−11	+	1.36E−07	+	4.50E−11	+
Cancer	9.60E−10	+	3.01E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
Balance	5.02E−11	+	0.01E−0	+	0.10E−0	−	8.09E−10	+	0.00E−0	+
Haberman	2.01E−11	+	1.07E−07	+	5.18E−07	+	1.11E−06	+	6.52E−07	+
Wine	5.16E−08	+	3.01E−11	+	3.02E−11	+	3.12E−11	+	3.12E−10	+

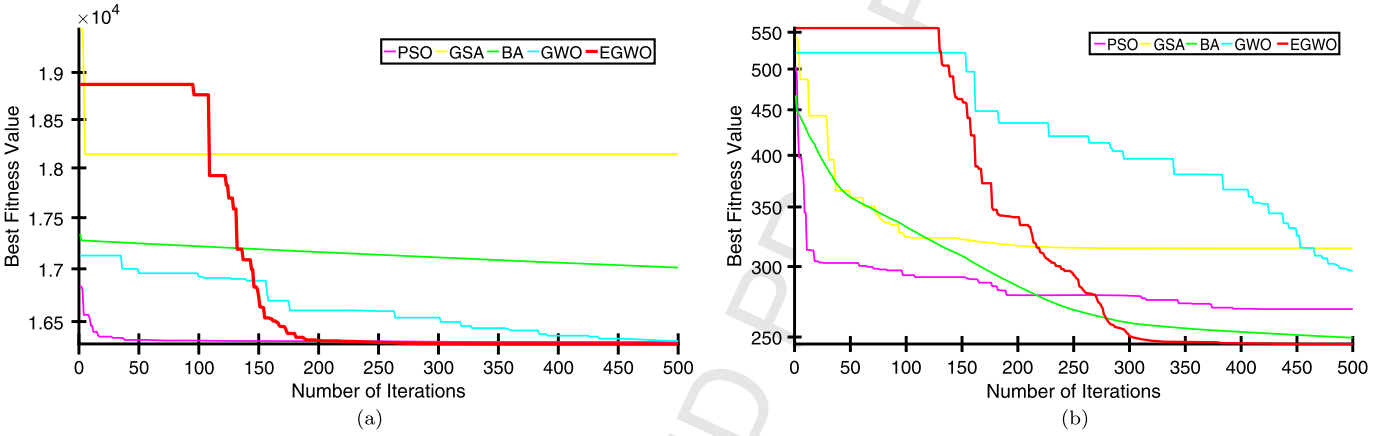

Fig. 2. The convergence graphs of (a) Wine and (b) Glass. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

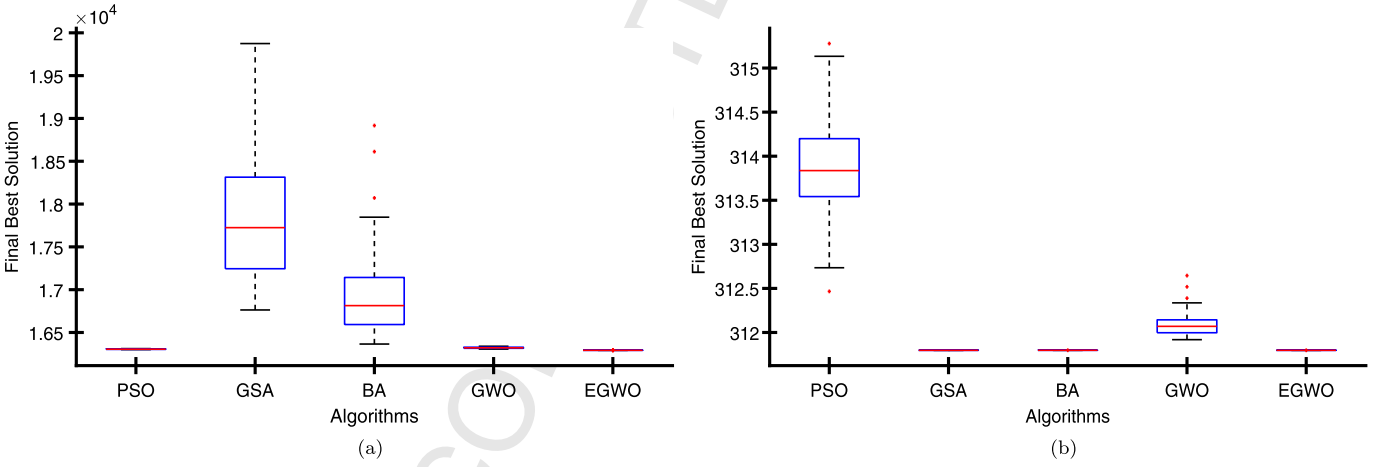

Fig. 3. The box-plot graphs for (a) Wine and (b) Glass.

results reported by the EGWO and other considered methods. Vertical lines of the boxes indicate variability of the best-so-far fitness value over 30 runs. Fig. 3a, 3b clearly illustrates that the degree of dispersion in EGWO is minimum, compared to PSO, GSA, BA, and GWO. Thus, it can be concluded from experimental analysis that EGWO is an efficient alternative for performing clustering tasks.

4.2. Performance analysis of MapReduce based EGWO (MR-EGWO)

In section 4.1, EGWO has shown to be an efficient alternative for clustering task. Thus, the performance of the parallelized EGWO, (MR-EGWO), is analyzed. Four large-scale synthetic datasets are used by duplicating each record of the original dataset 10^7 times. Table 5 briefs these datasets in terms of three parameters, namely, number of actual clusters ($\#C$), number of dimensions ($\#D$) and number of data-points ($\#N$). The required parameter setting of all the method is same as given in Table 2. For simulation, a Hadoop cluster of five nodes is designed where each node consists of an Intel Corei3-4570 processor with 3.20 GHz, 4 GB memory and 500 GB hard disk. Apache Hadoop version 2.6.2, java version 1.8.0 is used for the implementation of all methods and operating system is Ubuntu version 14.04. Table 6 shows the mean of F-measure and computation time for 4 large scale synthetic datasets obtained by running each method on a cluster of 5 machines. The F-measure comparison of four MapRedcue based methods as given

Table 5
Large datasets.

Dataset	#C	#D	#N
Replicated iris	3	7	10,000,050
Replicated CMC	3	9	10,000,197
Replicated wine	2	18	5000000
Replicated vovel	10	10	1025010

Please cite this article in press as: A.K. Tripathi et al., A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce, Big Data Res. (2018), https://doi.org/10.1016/j.bdr.2018.05.002

Table 6
Mean of F-measure and computation time over 30 runs.

S. No.	Dataset	Criteria	Parallel K-means	Parallel K-PSO	MR-ABC	DFBPKBA	MR-EGWO
1	Replicated iris	F-measure	0.667	0.785	0.842	0.790	**0.846**
		Computation time	**8.05E+04**	9.23E+04	9.26E+04	9.24E+04	9.22E+04
2	Replicated CMC	F-measure	0.298	0.324	0.387	0.378	**0.391**
		Computation time	**8.24E+E04**	10.33E+E04	10.33E+E04	10.34E+E04	10.32E+E04
3	Replicated wine	F-measure	0.482	0.517	0.718	0.719	**0.733**
		Computation time	**11.20E+04**	16.14E+04	16.23E+04	19.24E+04	16.11E+04
4	Replicated vovel	F-measure	0.586	0.627	0.634	0.622	**0.635**
		Computation time	**10.50E+04**	13.22E+04	12.21E+04	13.23E+04	13.21E+04

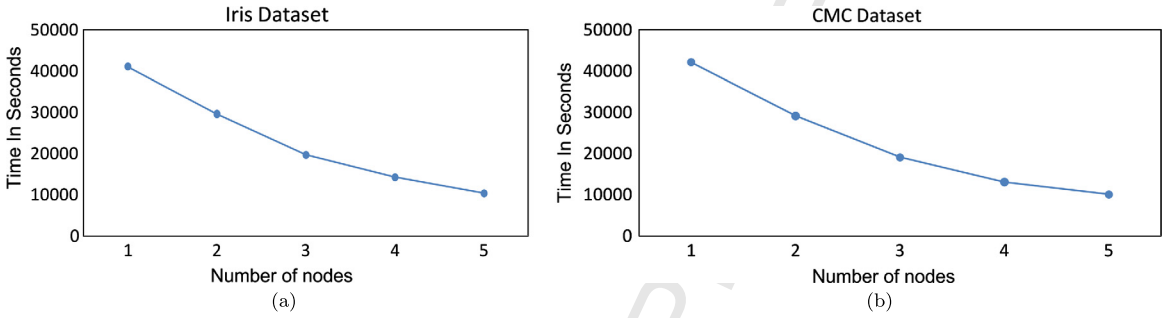

Fig. 4. The speedup graph of (a) Iris (b) CMC.

in Table 6 confirms that the proposed MR-EGWO outperformed all the methods under comparison while K-means has given the least performance among all the considered methods. However, the computation time of K-means is less as compared to the meta-heuristics based clustering methods. Thus, it can be concluded that the proposed method can be used for efficient clustering of large datasets.

Furthermore, the speedup performance of the MR-EGWO is analyzed on iris and CMC datasets. The speedup measure of a method is determined by Eq. (21).

$$S_p = T_{base}/T_N \tag{21}$$

where, T_{base} is the running time when p method runs on one machine and T_N is the running time of the same method runs on a cluster with N machines. To measure the speedup performance of MR-EGWO, one machine is increased in the cluster on each run. The speedup performance of MR-EGWO is illustrated in Fig. 4. It can be concluded from the Fig. 4 that the running time of MR-EGWO decreases gradually with the increase of machines in the Hadoop cluster. The proposed method has achieved up to 4.6754, 4.3457 speedup measure on dataset 1 and 2 respectively with the 5 machine cluster. Therefore, it is affirmed that the proposed MR-EGWO is advantageous for large-scale data.

5. Conclusion

In this paper, a novel MapReduce based clustering method is presented. The proposed method has three folds, (i) An efficient variant of grey-wolf optimizer called enhanced grey-wolf optimizer (EGWO) has been introduced for improving the quality of clustering (ii) The performance of the proposed variant (EGWO) is validated on seven benchmark datasets for clustering problem. The proposed method has outperformed five clustering methods namely: K-means, PSO, GSA, BA and GWO in terms of mean and best fitness values. The exploration and exploitation capabilities of the proposed variant are also analyzed using convergence graph. Boxplots are drawn to study the consistency of the results over the 30 runs. Third, a novel method named, MR-EGWO is proposed by parallelizing the EGWO using MapReduce for clustering large scale data sets. The proposed method, MR-EGWO, takes the advantage of EGWO to alleviate the clustering quality and MapReduce architecture to cope with large scale datasets.

Furthermore, to ascertain the efficiency of the MR-EGWO in the parallel environment, the proposed method is run on the Hadoop cluster of five nodes for four large scale synthetic datasets namely, iris, CMC, wine, and vovel. The simulation results outperformed four state-of-the-art MapReduce based clustering methods in terms of F-measure. Moreover, the speedup efficiency of the MR-EGWO is studied on two synthetic datasets (iris and CMC) by varying the number of nodes of the Hadoop cluster. The speedup results show that MR-EGWO is well suited for analyzing large datasets with significant speedup performance and better clustering quality. Thus, it is concluded that MR-EGWO is a competitive method for large scale clustering problems. In future, recent parallelization tools like spark may be tested to reduce the computation time of the proposed method. Moreover, the proposed method could be extended on some real-world clustering applications with large datasets like twitter analysis, video analysis, and satellite image analysis.

Uncited references

[26] [27] [28] [29] [30] [35] [36] [37] [44]

References

[1] U.M. Fayyad, A. Wierse, G.G. Grinstein, Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, 2002.
[2] M. Friedman, M. Last, Y. Makover, A. Kandel, Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology, Inf. Sci. 177 (2007) 467–475.
[3] L. Liao, T. Lin, B. Li, MRI brain image segmentation and bias field correction based on fast spatially constrained kernel clustering approach, Pattern Recognit. Lett. 29 (2008) 1580–1588.
[4] E.W. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Biometrics 21 (1965) 768–769.
[5] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.
[6] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, vol. 344, John Wiley & Sons, 2009.
[7] Y.-T. Kao, E. Zahara, I.-W. Kao, A hybridized approach to data clustering, Expert Syst. Appl. 34 (3) (2008) 1754–1762.
[8] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognit. 33 (9) (2000) 1455–1465.
[9] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (abc) algorithm, Appl. Soft Comput. 11 (1) (2011) 652–657.

Please cite this article in press as: A.K. Tripathi et al., A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce, Big Data Res. (2018), https://doi.org/10.1016/j.bdr.2018.05.002

[10] S. Alam, G. Dobbie, P. Riddle, Particle swarm optimization based clustering of web usage data, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, IEEE Computer Society, 2008, pp. 451–454.
[11] T. Ashish, S. Kapil, B. Manju, Parallel bat algorithm-based clustering using mapreduce, in: Networking Communication and Data Knowledge Engineering, Springer, 2018, pp. 73–82.
[12] V. Kumar, J.K. Chhabra, D. Kumar, Grey wolf algorithm-based clustering technique, J. Intell. Syst. 26 (1) (2017) 153–168.
[13] A. Hatamlou, S. Abdullah, H. Nezamabadi-Pour, A combined approach for clustering based on k-means and gravitational search algorithms, Swarm Evol. Comput. 6 (2012) 47–52.
[14] M. Ebrahimi, E. ShafieiBavani, R.K. Wong, S. Fong, J. Fiaidhi, An adaptive meta-heuristic search for the Internet of things, Future Gener. Comput. Syst. 76 (2017) 486–494.
[15] H. Mittal, M. Saraswat, An optimum multi-level image thresholding segmentation using non-local means 2d histogram and exponential kbest gravitational search algorithm, Eng. Appl. Artif. Intell. 71 (2018) 226–235.
[16] R. Pal, M. Saraswat, Data clustering using enhanced biogeography-based optimization, in: 2017 Tenth International Conference on Contemporary Computing, IC3, IEEE, 2017, pp. 1–6.
[17] A.C. Pandey, D.S. Rajpoot, M. Saraswat, Twitter sentiment analysis using hybrid cuckoo search method, Inf. Process. Manag. 53 (4) (2017) 764–779.
[18] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST, IEEE, 2010, pp. 1–10.
[19] Frontpage – hadoop wiki, http://wiki.apache.org/hadoop/. (Accessed 17 September 2016).
[20] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[21] S.N. Khezr, N.J. Navimipour, Mapreduce and its application in optimization algorithms: a comprehensive study, Majlesi J. Multimed. Process. 4 (3) (2015) 31–33.
[22] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[23] J. Wang, D. Yuan, M. Jiang, Parallel k-pso based on mapreduce, in: 2012 IEEE 14th International Conference on Communication Technology, ICCT, IEEE, 2012, pp. 1203–1208.
[24] C.-Y. Lin, Y.-M. Pai, K.-H. Tsai, C.H.-P. Wen, L.-C. Wang, Parallelizing modified cuckoo search on mapreduce architecture, J. Electron. Sci. Technol. 11 (2) (2013) 115–123.
[25] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, J.-J. Li, Distributed evolutionary algorithms and their models: a survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300.
[26] M.J. Meena, K. Chandran, A. Karthik, A.V. Samuel, An enhanced aco algorithm to select features for text categorization and its parallelization, Expert Syst. Appl. 39 (5) (2012) 5861–5871.
[27] I. Aljarah, S.A. Ludwig, 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, 2013, pp. 169–170.
[28] B. Wu, G. Wu, M. Yang, A mapreduce based ant colony optimization approach to combinatorial optimization problems, in: 2012 Eighth International Conference on Natural Computation, ICNC, IEEE, 2012, pp. 728–732.
[29] X. Xu, Z. Ji, F. Yuan, X. Liu, A novel parallel approach of cuckoo search using mapreduce, in: 2014 International Conference on Computer, Communications and Information Technology, CCIT 2014, Atlantis Press, 2014.
[30] V.S. Moertini, L. Venica, Enhancing parallel k-means using map reduce for discovering knowledge from big data, in: 2016 IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA, IEEE, 2016, pp. 81–87.
[31] A. Banharnsakun, A mapreduce-based artificial bee colony for large-scale data clustering, Pattern Recogn. Lett. 93 (2017) 78–84.
[32] V. Feoktistov, Differential Evolution: In Search of Solutions, vol. 5, Springer Science & Business Media, 2007.
[33] A.K. Tripathi, K. Sharma, M. Bala, Dynamic frequency based parallel k-bat algorithm for massive data clustering (DFBPKBA), Int. J. Syst. Assur. Eng. Manag. (2017) 1–9.
[34] W. Zhao, H. Ma, Q. He, Parallel k-means clustering based on mapreduce, in: IEEE International Conference on Cloud Computing, Springer, 2009, pp. 674–679.
[35] C.T. Brown, L.S. Liebovitch, R. Glendon, Lévy flights in dobe ju/hoansi foraging patterns, Hum. Ecol. 35 (1) (2007) 129–138.
[36] A.M. Reynolds, M.A. Frye, Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search, PLoS ONE 2 (4) (2007) e354.
[37] I. Pavlyukevich, Lévy flights, non-local search and simulated annealing, J. Comput. Phys. 226 (2) (2007) 1830–1844.
[38] S. Yang, R. Wu, M. Wang, L. Jiao, Evolutionary clustering based vector quantization and SPIHT coding for image compression, Pattern Recognit. Lett. 31 (13) (2010) 1773–1780.
[39] M.F. Shlesinger, G.M. Zaslavsky, U. Frisch, Lévy Flights and Related Topics in Physics, Lecture Notes in Physics, vol. 450, 1995, p. 52.
[40] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[41] S. Zhang, Y. Zhou, Grey wolf optimizer based on Powell local optimization method for clustering analysis, Discrete Dyn. Nat. Soc. (2015).
[42] X.-S. Yang, S. Deb, Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization, in: Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, 2010, pp. 101–111.
[43] C. Blake, C.J. Merz, {UCI} repository of machine learning databases.
[44] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, Appl. Soft Comput. 60 (2017) 115–134.
[45] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[46] T. Jayabarathi, T. Raghunathan, B. Adarsh, P.N. Suganthan, Economic dispatch using hybrid grey wolf optimizer, Energy 111 (2016) 630–641.
[47] S.A. Medjahed, T.A. Saadi, A. Benyettou, M. Ouali, Gray wolf optimizer for hyperspectral band selection, Appl. Soft Comput. 40 (2016) 178–186.
[48] N. Jayakumar, S. Subramanian, S. Ganesan, E. Elanchezhian, Grey wolf optimization for combined heat and power dispatch with cogeneration systems, Int. J. Electr. Power Energy Syst. 74 (2016) 252–264.
[49] G. Komaki, V. Kayvanfar, Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time, J. Comput. Sci. 8 (2015) 109–120.
[50] D. Guha, P.K. Roy, S. Banerjee, Load frequency control of interconnected power system using grey wolf optimization, Swarm Evol. Comput. 27 (2016) 97–115.
[51] S. Amirsadri, S.J. Mousavirad, H. Ebrahimpour-Komleh, A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training, Neural Comput. Appl. (2017) 1–14.
[52] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, W. Cai, Grey wolf optimizer for parameter estimation in surface waves, Soil Dyn. Earthq. Eng. 75 (2015) 147–157.
[53] S. Mirjalili, How effective is the grey wolf optimizer in training multi-layer perceptrons, Appl. Intell. 43 (1) (2015) 150–161.
[54] Y.-C. Ho, D.L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, J. Optim. Theory Appl. 115 (3) (2002) 549–570.
[55] A.A. El-Fergany, H.M. Hasanien, Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms, Electr. Power Compon. Syst. 43 (13) (2015) 1548–1559.
[56] M. Shakarami, I.F. Davoudkhani, Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay, Electr. Power Syst. Res. 133 (2016) 149–159.

Please cite this article in press as: A.K. Tripathi et al., A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce, Big Data Res. (2018), https://doi.org/10.1016/j.bdr.2018.05.002