

# Determining Number of Clusters Using Firefly Algorithm with Cluster Merging for Text Clustering

Athraa Jasim Mohammed<sup>1,2(✉)</sup>, Yuhanis Yusof<sup>1</sup>, and Husniza Husni<sup>1</sup>

<sup>1</sup> School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

s94734@student.uum.edu.my, autoathraa@yahoo.com,  
{yuhanis, husniza}@uum.edu.my

<sup>2</sup> University of Technology, Baghdad, Iraq

**Abstract.** Text mining, in particular the clustering is mostly used by search engines to increase the recall and precision of a search query. The content of online websites (text, blogs, chats, news, etc.) are dynamically updated, nevertheless relevant information on the changes made are not present. Such a scenario requires a dynamic text clustering method that operates without initial knowledge on a data collection. In this paper, a dynamic text clustering that utilizes Firefly algorithm is introduced. The proposed, aFA<sub>merge</sub>, clustering algorithm automatically groups text documents into the appropriate number of clusters based on the behavior of firefly and cluster merging process. Experiments utilizing the proposed aFA<sub>merge</sub> were conducted on two datasets; 20Newsgroups and Reuter's news collection. Results indicate that the aFA<sub>merge</sub> generates a more robust and compact clusters than the ones produced by Bisect K-means and practical General Stochastic Clustering Method (pGSCM).

**Keywords:** Firefly algorithm · Text clustering · Text mining · Agglomerative clustering

## 1 Introduction

Text clustering technique is widely applied in Information Retrieval (IR) (i.e. search engine) to enhance and improve the retrieval process [1]. Text clustering classifies a collection of documents clusters, where documents with high similarity between them is located in one cluster and the ones with less similarity are in another cluster [2]. Based on literatures [3, 4], text clustering methods can be divided into two main categories; partitional clustering and hierarchical clustering. The first category, groups a set of documents into flat clustering based on pre-defined criteria such as the one implemented in K-means [6]. On the other side, hierarchical clustering classifies a collection into hierarchical structure such as shown in Bisect K-means [5]. Based on previous studies [5, 6], it is learned that hierarchical clustering is better than partitional clustering in generating quality results. However, the Bisect k-means requires initial information about the dataset such as the number of clusters. Such information which must be provided by the dataset owner may be not available in some situation, hence indicating

the need for an alternative clustering algorithm that does not rely on initial description of a collection that is under analysis.

In literatures [1, 3, 7], some researchers try to overcome this problem by two approaches; estimation [1] and swarm [3, 7]. The first approach utilizes performance metrics to identify the number of cluster but this approach relies on the determination of the upper and lower value of a metric's range. The second approach is more preferable as it works dynamically. In this paper, a variant of Firefly algorithm text clustering, denoted as aFA<sub>merge</sub>, is proposed. This algorithm has ability the to automatically determine the optimal number of clusters without any initial knowledge about a collection (e.g. number of clusters). Further, the aFA<sub>merge</sub> employs a new procedure in merging the obtained clusters and this is based on a threshold. The proposed merging algorithm is believed to be dynamic as it clusters any given collection into the optimal number of clusters. The rest of the paper is organized as follows; Sect. 2 includes a discussion on related work. Section 3 contains elaboration on the proposed aFA<sub>merge</sub> clustering algorithm while Sect. 4 includes the experimental results. Finally, the conclusion of the work is presented in Sect. 5.

## 2 Related Work

Hierarchical clustering is an efficient technique in information retrieval as it can be used to construct a hierarchy of nested clusters [4, 8]. Existing Hierarchical clustering methods can be classified into two categorized; divisive and Agglomerative clustering algorithms [9, 10]. Divisive clustering algorithms operate with a single large cluster and divide it until the desired number of clusters is generated or a stopping condition is met. Its final outcome will be a tree of clusters. The Bisect K-means [4, 9] is a familiar method of divisive clustering, where at each level, Bisect K-means utilizes K-means [5] to split a cluster into two clusters. The process of splitting continues until it obtains  $k$  number of clusters.

On the other hand, the agglomerative clustering algorithms start with multi clusters, where each cluster includes one or more documents. These clusters later undergo a merging process based on chosen linkage metric. There are three linkage metrics; The Single Linkage Hierarchical Clustering (SLHC), Complete Linkage Hierarchical Clustering (CLHC) and Average Linkage Hierarchical Clustering (ALHC) [11]. The Single Linkage Hierarchical Clustering (SLHC) is a simple agglomerative clustering that measures similarity between two clusters by the closest pair of data objects. SLHC merges two clusters that have high similarity or have least amount of distance between closest pair. It is susceptible in dealing with noise and outliers [12]. On the other hand, the Complete Linkage Hierarchical Clustering (CLHC) measures the similarity between two clusters by farthest pair of data objects. CLHC merges two clusters that have minimum similarity or have maximum distance between pair of objects which is reverse of SLHC [13]. CLHC is less susceptible to noise and outliers, however it can break large groups and prefer spherical shapes [12]. The Average Linkage Hierarchical Clustering (ALHC) is one of the most popular agglomerative clustering to merge two clusters [10]. An example of ALHC is Un-weighted Pair Group Method with Arithmetic Mean

(UPGMA) [4, 10] which is based on the group average similarity among all objects in two clusters. The advantage of this method is that it can transact with dynamic data sets and prevents overlapping clustering [14]. However, the UPGMA consume high computational time [4]. The formula of merging two clusters as in UPGMA is based on cosine similarity and is shown in Eq. 1.

$$UPGMA_{i,j} = \frac{\sum_{i \in C} \sum_{j \in C} \text{Similarity}(D_i, D_j)}{N_i N_j} \quad (1)$$

where,  $N_i$  is the number of documents in cluster  $i$ , and  $N_j$  is the number of documents in cluster  $j$ .

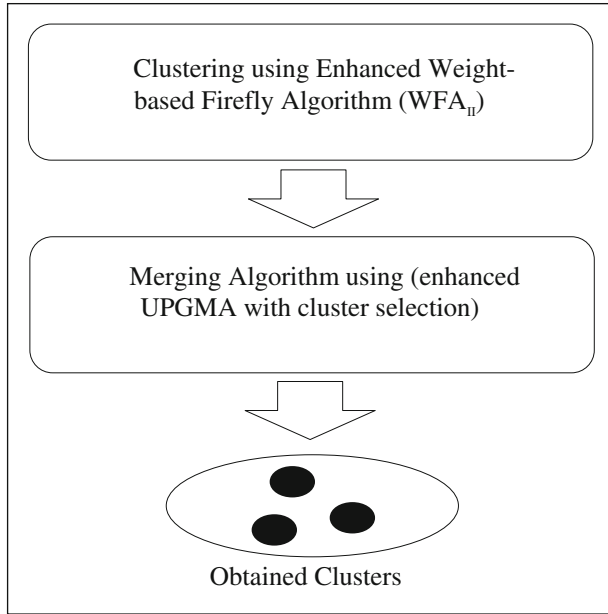
In [10], Yujian and Liye presented an improved Un-weighted Multiple Group Method with Arithmetic Mean (UMGMA) to solve the problem of tie trees (two or more trees create from analysing related populations) in UPGMA. The result enhances the UPGMA and produces a unique tree. However, this process also requires high computational time. In [4], Murugesan and Zhang proposed to utilize UPGMA to refine clusters generated by Bisect K-means and to reduce the time complexity of UPGMA. The result of the proposed method outperformed bisect K-means in three performance metrics; however, such approach relies on a predefine value (i.e. number of clusters).

In some other related work [7, 15, 16], clustering methods using swarm intelligence algorithm were introduced. Swarm based clustering has the ability to automatically determine the optimal or near optimal number of clusters. Swarm based clustering adapts the behavior of a specific insect or animal in the nature and converts it to heuristics rules. Flocking based approach [15] relates to the behavior of group of flocks, while the Ant based clustering [16] operates based on the behavior of ants. In the work of Tan et al. [7], the practical General Stochastic Clustering Method (pGSCM) that simplifies the ant based clustering was introduced on multivariate real world data.

Following the ant approach, another known swarm algorithm is the Firefly Algorithm (FA) which was introduced by Yang in 2010 [17, 18]. FA relates with the behavior of firefly insect that can automatic subdivide division into subgroups and offers the capability of multi-modality. FA has two major factors; the flashing light that is indicates fitness of a firefly and the attractiveness between fireflies that represent the distance between two fireflies. FA was presented to solve various optimization problems, including economic dispatch problems [24], anomaly detection [25], and data clustering [26, 27]. As the utilization of FA has proven to be successful, this study proposes hierarchical text clustering that is based on Firefly algorithm.

### 3 Proposed Clustering Method

Figure 1 presents the main steps in the proposed aFA<sub>merge</sub> clustering; document clustering using Enhanced Weight-based Firefly algorithm [19] and cluster merging using an enhanced UPGMAE with cluster selection.



**Fig. 1.** Two main steps in  $aFA_{merge}$ .

### 3.1 Clustering using Enhanced Weight-based Firefly Algorithm II (WFA<sub>II</sub>)

Referring to the work presented in WFAII [19], the number of fireflies used in the clustering process equals the number of documents. Initially, the weight of each document is assigned to each firefly as its initial light as shown in step 2 in Fig. 2. Fireflies compete between each other based on the brightness of light and similarity between them as illustrated in steps 9 and 10 in Fig. 2. If two fireflies are successful, the one with a bright light will attract the less bright one and the position of the firefly with the less bright light will be changed, where it is moved near to the winning firefly. After some number of iteration (in this study, it is set to 20), all fireflies are ranked based on their brightness. Firefly with the brightness light is chosen as center of cluster. Relevant documents are assigned to this first cluster, while the remaining documents (fireflies) are ranked again to produce a new center and new clusters. This process continues until the last document is clustered. The number of clusters obtained via the clustering process is usually large; hence there is a need to merge the produced clusters.

### 3.2 Merging Algorithm

The obtained clusters are later merged using the Enhanced Un-weighted Pair Group Method with Arithmetic Mean (UPGMA<sub>E</sub>) and cluster selection. The required process in the proposed UPGMA<sub>E</sub> is illustrated in the following steps:

Step1: Generate Initial population of firefly  $x_i$  where  $i=1, 2, \dots, n$ ,  $n$ =number of fireflies (documents).  
 Step2: Initial Light Intensity,  $I$ =total weight of document.  
 Step3: Define light absorption coefficient  $\gamma$ , initial  $\gamma=1$ .  
 Step4: Define the randomization parameter  $\alpha$ ,  $\alpha=0.2$ .  
 Step5: Define initial attractiveness  $\beta_0=1.0$ .  
 Step6: While  $t < \text{Number of iteration}$   
 Step7: For  $i=1$  to  $N$   
 Step8: For  $j=1$  to  $N$   
 Step9: If (total weight  $I_i < \text{total weight } I_j$ ) {  
 Step10: If Similarity ( $i, j$ )  $\geq \text{Threshold}$  {  
 Step11: Calculate distance between  $i, j$  using equation below:  

$$\text{CartesianDistance}(X_i, X_j) = \sqrt[2]{(X_i - X_j)^2}^{(-\gamma r_{ij}^2)}$$
  
 Step12: Calculate attractiveness using equation:  $\beta = \beta_0 \exp$   
 Step13: Move document  $i$  to  $j$  using equation:  $X^i = X^i + \beta * (X^j - X^i) + \alpha \epsilon_i$   
 Step14: Update light intensity ( $I$ ) using equation:  $I(d_j) = I(d_j) + \beta$   
 Step15: End For  $j$   
 Step16: End For  $i$   
 Step17: Loop  
 Step18: Rank to find best document.

**Fig. 2.** Pseudo code of weight-based firefly algorithm II (WFA<sub>II</sub>) [19]

- Step 1: Check whether to merge the first cluster with other clusters of the produced clusters (i.e. output clusters). If no merging is required, eliminate the first cluster from the output clusters (meaning that the cluster is not included in merging process), then the second cluster becomes the first cluster. The process of step 2–7 continues until the last cluster becomes the first cluster.
- Step 2: Suppose that  $C_1$  and  $C_2$  are two clusters that are to be merged, and suppose that  $P_1$  and  $P_2$  are the number of documents in the two clusters respectively.
- Step 3: Suppose that CSim is the Cosine similarity matrix between the two clusters,  $C_1$  and  $C_2$ . The documents in  $C_1$  is represented as row and the documents in  $C_2$  is represented as column. The value of CSim matrix is equal to 1 if the document in  $C_1$  is similar to document in  $C_2$ , else is equal to 0. The similarity between two documents is based on threshold.
- Step 4: If (number of documents in cluster  $C_1 \geq 2$  and number of documents in cluster  $C_2 \geq 2$ ) OR (If number of documents in cluster  $C_1 \geq 3$  and number of documents in cluster  $C_2 == 1$ ) OR (If number of documents in cluster  $C_2 \geq 3$  and number of documents in cluster  $C_1 == 1$ ) then

Step 5: Calculate the average similarity between two clusters as in Eq. 2.

$$\frac{1}{P_1} \sum_{i=1}^{P_1} \sum_{j=1}^{P_2} \frac{CSim(C_i, C_j)}{P_2} \quad (2)$$

Where,  $P_1$  is the number of document in the first cluster,  $P_2$  is the number of document in second cluster,  $C_i$  is the first cluster and  $C_j$  is the second cluster.

Step 6: Calculate the merge threshold as in Eq. 3 below.

$$MergeThreshold(MT) = floor \left( \frac{round \left( \frac{P_1 * P_2}{2} \right) - 1}{P_1 * P_2} * 10 \right) / 10 \quad (3)$$

Step 7: If the value of Eq. 2 is larger than the *MergeThreshold* in Eq. 3, as shown in Eq. 4, then combine the two clusters  $C_1$  and  $C_2$  in one cluster.

$$\frac{1}{P_1} \sum_{i=1}^{P_1} \sum_{j=1}^{P_2} \frac{CSim(C_i, C_j)}{P_2} \geq MergeThreshold(MT) \quad (4)$$

Step 8: If (number of documents in cluster  $C_1 \geq 2$  and number of documents in cluster  $C_2 \geq 1$ ) OR If (number of documents in cluster  $C_2 \geq 2$  and number of documents in cluster  $C_1 \geq 1$ )

Step 9: Combine  $C_1$  and  $C_2$ , if Eq. 5 is true

$$MergeThreshold(MT) = \frac{round \left( \frac{P_1 * P_2}{2} \right)}{P_1 * P_2} \quad (5)$$

Step 10: If (number of documents in cluster  $C_1 \geq 1$  and number of documents in cluster  $C_2 \geq 1$ )

Step 11: Combine  $C_1$  and  $C_2$ , if  $CSim(C_1, C_2)$  equals 1.

Clusters selection operates once the merge using  $UPGMA_E$  is performed. The merging may produce pure clusters but they are of different sizes (big and small size of clusters). Hence, there is a need to select the big size clusters (the pure ones with large number of documents) and merge them with small size clusters (clusters with small number of documents). The cluster selection process chooses clusters that exceed an identified threshold which is (50,  $n/20$ ) as adopted from [15, 16]. The required steps in cluster selection is shown in the following.

- Step 1: Set selected threshold equal to  $\min(50, n/20)$ , where  $n$  is the total number of documents.
- Step 2: For  $i = 1$  and until the number of produced clusters
- Step 3: If  $\text{length}(C_i) \geq \min(50, n/20)$
- Step 4: Save  $C_i$  in selected clusters (Big size clusters)
- Step 5: Else Save  $C_i$  in non-selected clusters (small size clusters)
- Step 6: End.

## 4 Experimental Results

In order to evaluate the proposed  $\text{aFA}_{\text{merge}}$ , an experiment was conducted on two datasets; the 20Newsgrroups [20] and Reuters [21]. Table 1 depicts a summary on the datasets.

**Table 1.** Characteristics of datasets

| Datasets      | Total of documents | Number of classes | Number of terms |
|---------------|--------------------|-------------------|-----------------|
| 20Newsgrroups | 300                | 3                 | 2275            |
| Reuters       | 300                | 6                 | 1212            |

The evaluation of  $\text{aFA}_{\text{merge}}$  is performed based on three external metrics mostly used in text clustering. These metrics are Purity, F-measure and Entropy [4]. Results obtained by the proposed  $\text{aFA}_{\text{merge}}$  is compared against the ones produced by two types of clustering methods; static method such as Bisect K-means [4, 9] and Dynamic method such as pGSCM [7]. All experiments were carried out in Matlab on windows 8 with a 2000 MHz processor and 4 GB memory. The execution of  $\text{aFA}_{\text{merge}}$ , Bisect K-means [4, 9] and pGSCM [7] is of ten (10) times and the average value of the metrics were recorded.

### 4.1 Results and Discussion

Table 2 tabularizes the experimental results of Purity, F-measure, and Entropy for the three clustering;  $\text{aFA}_{\text{merge}}$ , Bisect K-means [4, 9] and pGSCM [7]. As can be seen from data in Table 2, the purity value for  $\text{aFA}_{\text{merge}}$  is higher than Bisect K-means and pGSCM in both datasets; 20Nwesgrroups and Reuters. The highest purity value are (0.4920) and (0.6217) while Bisect K-means produces (0.4303) and (0.4150), and pGSCM generates (0.3853) and (0.2357). The illustrative Purity results of the proposed  $\text{aFA}_{\text{merge}}$ , Bisect K-means and pGSCM is shown in Fig. 3(a). On the other hand, despite of  $\text{aFA}_{\text{merge}}$  generating the highest F-measure in the Reuters dataset, it was defeated by Bisect K-means for the 20Newsgrroups dataset. This is shown in Fig. 3(b). Further, it is noted from Table 2 that the Entropy of the proposed  $\text{aFA}_{\text{merge}}$  is less than Bisect K-means and

pGSCM in both datasets, where the best value of proposed  $aFA_{merge}$  is (1.3319) while Bisect K-means generates (1.4621) and (1.8839), and pGSCM generates (1. 5630) and (2.5144). The Entropy results of  $aFA_{merge}$ , Bisect K-means and pGSCM is illustrated in Fig. 3(c). It is learned from the literature that high value of purity and F-measure (i.e. value approaching 1) and small Entropy value (approaching to 0) indicates a better clustering (quality clusters) [4, 22].

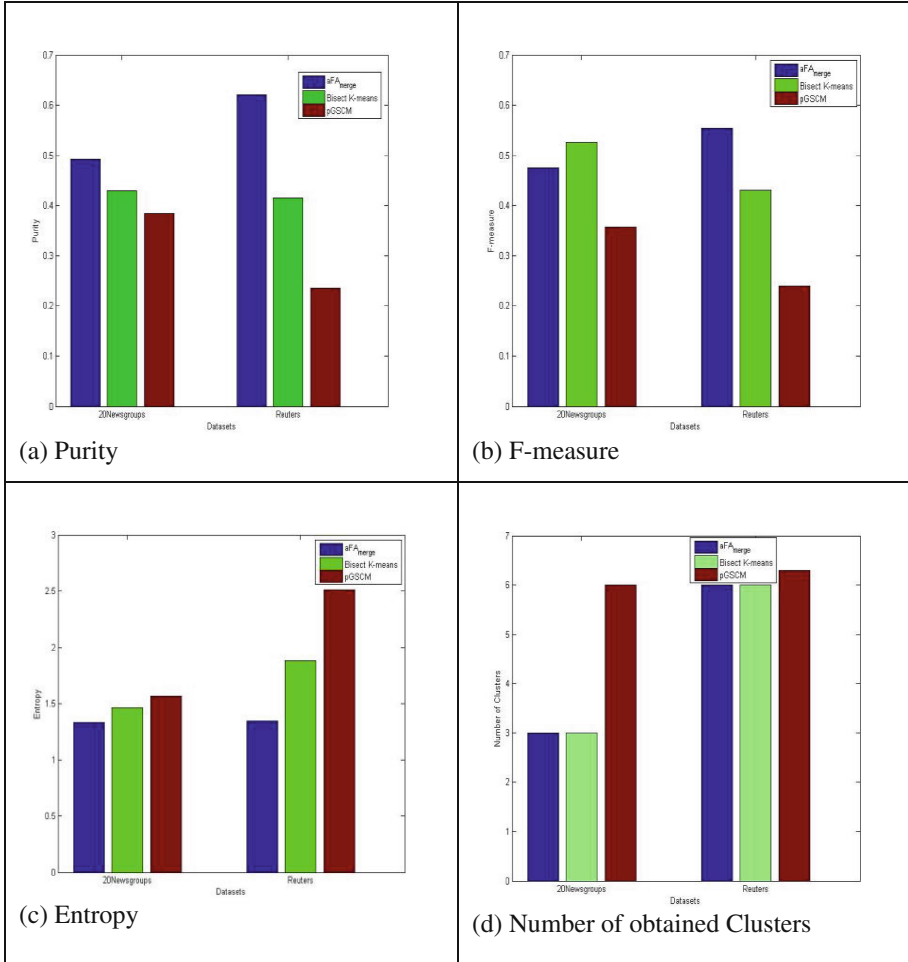
**Table 2.** Clustering results:  $aFA_{merge}$  vs. bisect K-means vs. pGSCM.

| Datasets      | Algorithms     | Performance metrics              |                                  |                                  | # Number of clusters |
|---------------|----------------|----------------------------------|----------------------------------|----------------------------------|----------------------|
|               |                | Purity                           | F-measure                        | Entropy                          |                      |
| 20News-groups | $aFA_{merge}$  | <b>0.4920</b><br>(0.0190)        | 0.4764<br><b>(0.0032)</b>        | <b>1.3319</b><br>(0.0335)        | <b>3</b>             |
|               | Bisect K-means | 0.4303<br>(0.1150)               | <b>0.5264</b><br>(0.0717)        | 1.4621<br>(0.1875)               | 3                    |
|               | pGSCM          | 0.3853<br><b>(0.0159)</b>        | 0.3571<br>(0.0275)               | 1.5630<br><b>(0.0132)</b>        | 6                    |
| Reuters       | $aFA_{merge}$  | <b>0.6217</b><br><b>(0.0042)</b> | <b>0.5538</b><br><b>(0.0024)</b> | <b>1.3414</b><br><b>(0.0154)</b> | <b>6</b>             |
|               | Bisect K-means | 0.4150<br>(0.1022)               | 0.4307<br>(0.1142)               | 1.8839<br>(0.3037)               | 6                    |
|               | pGSCM          | 0.2357<br>(0.0147)               | 0.2400<br>(0.0108)               | 2.5144<br>(0.0245)               | <b>6.3</b>           |

*Note: highlighted value in ‘bold’ ‘indicates the best value while the standard deviation is included in ().*

As can be observed in Table 2, the number of obtained clusters by the proposed  $aFA_{merge}$  is (3) for 20Newsgroups dataset and (6) for Reuters dataset. Such a result is the same as the original classes. On the other hand, pGSCM generates double the number of clusters (i.e. 6) for 20Newsgroups dataset. Figure 3(d) shows the graphical representation of the obtained results on number of clusters. In addition, it can see from Table 2, in most metrics, the standard deviation of solution found by proposed  $aFA_{merge}$  is smaller than the ones obtained using the other two methods; Bisect K-means and pGSCM. As noted from [23], the lower the value of standard deviation for solution generated by a specific method, the better the solution is. Hence, the obtained clustering results indicate that the proposed  $aFA_{merge}$  is a better clustering than the chosen benchmark methods.





**Fig. 3.** Cluster quality results and number of clusters: aFA<sub>merge</sub> vs. bisect K-means vs. pGSCM.

## 5 Conclusion

In this study, a dynamic text clustering method based on Firefly algorithm is proposed. The aFA<sub>merge</sub> uses fireflies to automatically classify text documents without the need of prior knowledge on the data collection. The obtained clusters were then refined using the Enhanced Un-weighted Pair Group Method with Arithmetic Mean (UPGMA<sub>E</sub>) and cluster selection. The proposed aFA<sub>merge</sub> is realized on benchmark datasets which includes 20Newsgroups and Reuters news. Experimental results demonstrate that aFA<sub>merge</sub> clustering produces better clusters than Bisect K-means and pGSCM. Furthermore, aFA<sub>merge</sub> clustering produced the exact number of clusters as occurred in the actual

grouping. Such a result indicates that the proposed  $aFA_{\text{merge}}$  would be useful for a search engine in presenting users with the required retrieval.

**Acknowledgments.** Authors would like to thank the Malaysian Ministry of Higher Education for providing the financial support under the Fundamental Research Grant Scheme (s/o: 12894). Gratitude also goes to Universiti Utara Malaysia for helping in managing the study.

## References

1. Sayed, A., Hacid, H., Zighed, D.: Exploring validity indices for clustering textual data. In: Zighed, D.A., Tsumoto, S., Ras, Z.W., Hacid, H. (eds.) *Mining Complex Data. Studies in Computational Intelligence*, vol. 165, pp. 281–300. Springer, Heidelberg (2009)
2. Miner, G., Elder, J., Fast, A., Hill, T., Nisbet, R., Delen, D.: *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*, 1st edn. Elsevier, Amsterdam (2012)
3. Zhang, L., Cao, Q., Lee, J.: A novel ant-based clustering algorithm using Renyi entropy. *Appl. Soft Comput.* **13**(5), 2643–2657 (2013)
4. Murugesan, K., Zhang, J.: Hybrid bisect K-means clustering algorithm. In: *IEEE International Conference on Business Computing and Global Informatization (BCGIN)*, pp. 216–219. IEEE (2011)
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323 (1999)
6. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: *proceedings of KDD Workshop on Text Mining*, Boston (2000)
7. Tan, S.C., Ting, K.M., Teng, S.W.: A general stochastic clustering method for automatic cluster discovery. *Pattern Recogn.* **44**(10–11), 2786–2799 (2011)
8. Feng, L., Qiu, M.H., Wang, Y.X., Xiang, Q.L., Yang, Y.F.: Fast divisive clustering algorithm using an improved discrete particle swarm optimizer. *Pattern Recogn. Lett.* **31**, 1216–1225 (2010)
9. Kashef, R., Kamel, M.S.: Enhanced bisecting K-means clustering using intermediate cooperation. *Pattern Recogn.* **42**(11), 2557–2569 (2009)
10. Yujian, L., Liye, X.: Unweighted multiple group method with arithmetic mean. In: *the IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pp. 830–834 (2010)
11. Yin, Y., Kaku, I., Tang, J., Zhu, J.: *Data Mining Concepts, Methods and Application in Management and Engineering Design*. Springer, London (2011)
12. Tan, P.N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*. Pearson Education, New York, Addition Wesley, Boston (2006)
13. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*, 1st edn. Cambridge University Press, Cambridge (2008)
14. Gil-Garcia, R., Pons-Porrata, A.: Dynamic hierarchical algorithms for document clustering. *Pattern Recogn. Lett.* **31**(6), 469–477 (2010)
15. Picarougne, F., Azzag, H., Venturini, G., Guinot, C.: A new approach of data clustering using a flock of agents. *Evol. Comput.* **15**(3), 345–367 (2007)
16. Tan, S.C., Ting, K.M., Teng, S.W.: Simplifying and improving ant-based clustering. *Procedia Comput. Sci.* **4**, 46–55 (2011)
17. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2**(2), 78–84 (2010)

18. Yang, X.S., He, X.: Firefly algorithm: recent advances and applications. *Int. J. Swarm Intell.* **1**(1), 36–50 (2013)
19. Mohammed, A.J., Yusof, Y., Husni, H.: Document clustering based on firefly algorithm. *J. Comput. Sci.* **11**(3), 453–465 (2015)
20. Newsgroup Data Set (2006). <http://people.csail.mit.edu/20Newsgroup/>
21. Lewis, D.: The reuters-21578 text categorization test collection (1999). <http://kdd.ics.uci.edu/database/reuters21578/reuters21578.html>
22. Forsati, R., Mahdavi, M., Shamsfard, M., Meybodi, M.R.: Efficient stochastic algorithms for document clustering. *Inf. Sci.* **220**, 269–291 (2013)
23. Hatamlou, A., Abdullah, S., Nezamabadi-pour, H.: A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm Evol. Comput.* **6**, 47–52 (2012)
24. Yang, X.S., Hosseini, S.S.S., Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl. Soft Comput.* **12**(3), 1180–1186 (2012)
25. Adaniya, M.H.A.C., Abrão, T., Proença Jr., M.L.: Anomaly detection using metaheuristic firefly harmonic clustering. *J. Netw.* **8**(1), 82–91 (2013)
26. Banati, H., Bajaj, M.: Performance analysis of firefly algorithm for data clustering. *Int. J. Swarm Intell.* **1**(1), 19–35 (2013)
27. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study. *Swarm Evol. Comput.* **1**(3), 164–171 (2011)

Advances in Visual Informatics

4th International Visual Informatics Conference, IVIC

2015, Bangi, Malaysia, November 17-19, 2015,

Proceedings

Badioze Zaman, H.; Robinson, P.; Smeaton, A.F.; Shih,

T.K.; Velastin, S.; Jaafar, A.; Mohamad Ali, N. (Eds.)

2015, XV, 526 p. 213 illus. in color., Softcover

ISBN: 978-3-319-25938-3