International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 7 (2015) pp. 17429-17440

http://www.ripublication.com

An Effective Algorithm For Web Search Result Clustering Based on Cuckoo Search and Artificial Bee Colony Algorithms

P.Salini*, P.Umaa Shangari, G.A Nagalakshmi, K.Sangeetha Kumari
Department of Computer Science Engineering,
Pondicherry Engineering College
Puducherry, India.
**salini@pec.edu*

Abstract

Clustering of web search results or web document clustering has become an interesting research area in information retrieval. Web Clustering Engines increases the number of relevant documents for the user to review, while reducing the time spent on reviewing them. There are several algorithms used for clustering namely suffix tree clustering, k-means, LINGO but it also provides a room for more. In this paper, a new description centric algorithm, called web search result clustering is proposed using Cuckoo search meta-heuristic algorithm and Artificial Bee Colony bio-inspired algorithm. The cuckoo search provides a combined global and local search strategy in solution space. The proposed algorithm was tested with the benchmark datasets over many queries. The algorithm was also compared against other web search result clustering algorithms namely suffix tree clustering, lingo, k-means and web search result clustering. The following parameters namely precision, recall, F-measure, accuracy and SSL_{k} were used to evaluate the proposed algorithm.

Introduction

In recent years, Web Search Result Clustering has become a very interesting research area among academic and scientific communities involved in Information Retrieval (IR) and web search [3]. This is because, it is most likely that results relevant to the user are close to each other in the document space, thus it falls into a relatively small number of clusters [4] and thereby reducing the search time. These web search result clustering systems are called as web clustering engines and the main exponents in the field are Carrot2 (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, which is known as Vivisimo), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it). The clustering systems usually consists of four main

components: (i) Retrieval of search results, (ii) pre-processor, (iii)cluster formation and labelling, and (iv) visualization of clusters [2] (see Fig. 1).

The retrieval of search results begins with a query defined by the user and based on the user's query, relevant documents search is conducted in diverse data sources. The web clustering engines work as meta-search engines. It collects 50 to 200 results from traditional search engines. These results will contain a URL, snippet and title of each document [2].

The Pre-processor will convert each of the snippets into a sequence of words or general attributes or characteristics, so that it can be used as the input to clustering algorithm. The Pre-processor performs the tasks such as removing the special characters and accents, converting the strings to lowercase, stop word removal and stemming of words [2].

The Pre-processed snippets (features) are then fed into the clustering formation and labelling component. This component makes use of any of clustering algorithms [2]: types namely data centric, description aware and description centric which helps to build clusters of documents and assign an appropriate label to each of the clustered group.

Finally, in the visualization component engine, the clustering engine will display the results to the user in hierarchically organized folders. The folders will have label or title that represents the documents that it contains such that it can be easily identified by the user.

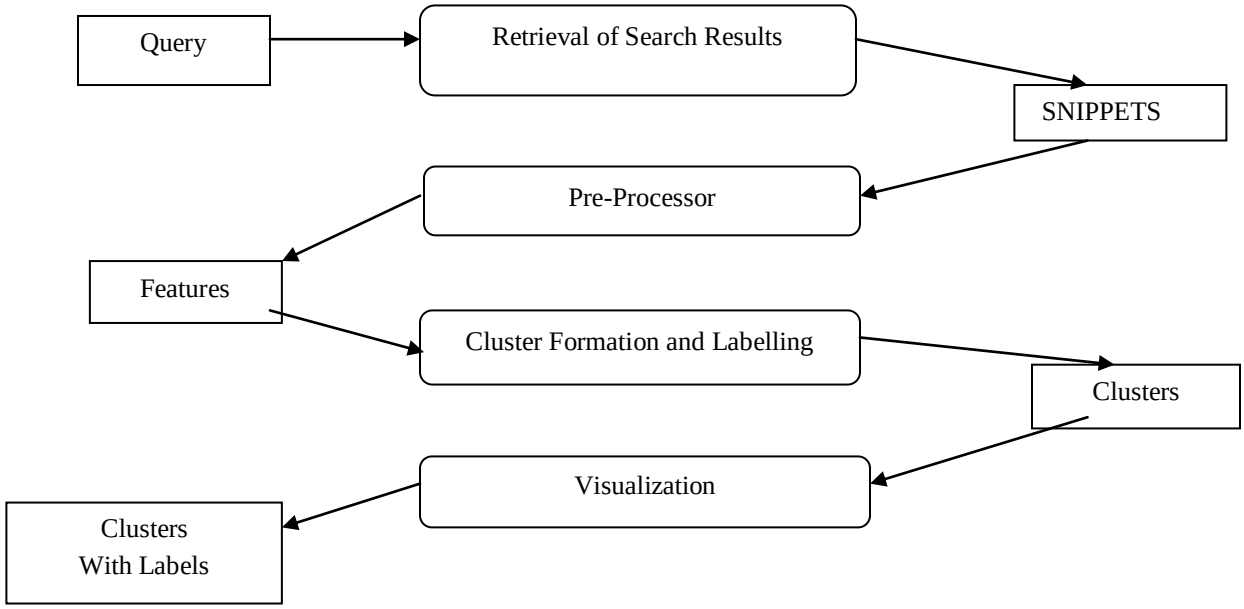

Figure 1: Components of Web Search Results Clustering Engine

The web document clustering algorithms must meet the following requirements : (i) automatically define the number of clusters to be created; (ii) generate relevant clusters the appropriate documents clusters; (iii) define labels or names for the

clusters for easy understanding of users; (iv) handle overlapping clusters (i.e. documents can belong to more than one cluster); (v) reduce the high dimension of document collections; (vi) handle the processing time, (i.e. less than or equal to 2s); and (vii) handle the noise found in documents [1].

In the existing systems, the processing time is not handled and retrieve high dimension of document collections. As a solution the proposed system named Web Search Result Clustering based on Cuckoo Search and Artificial Bee Colony Algorithms (WSRC-CSABC) for clustering WSRC with all the requirements and proper handling of processing time and high dimension clusters.

Related Works

In this section, some of the works related to web search result clustering under each type is discussed followed by the cuckoo search algorithm.

Type of Clustering Algorithms

The first type of clustering algorithm is Data centric algorithm that is used for data clustering [5] which are very strong, but has a flaw in the presentation of the labels or in the explanation of the groups. The algorithms most commonly used for clustering of web search results are been the hierarchical and the partitioned [6].

A link-based algorithm was proposed in 2009 [13] that make use of the hyperlink structure to find dense unit and it also improves the joining process for creating hierarchical clusters of web documents. The advantage of this algorithm is that it can create clusters of different shape and can also remove the noisy data. For joining process, it uses a measure that dynamically determines the cluster boundaries. When compared against other density-based clustering algorithm this result show a higher clustering quality. This algorithm gives less attention cluster labels.

A new learning algorithm based on k-means and neural networks was also proposed in 2009 [14]. This uses Principal Component Analysis (PCA) in order to reduce the dimensionality of the document matrix, singular value decomposition to find the measure of similarity and the multilayer neural network for reducing the time of the document clustering process. The algorithm did not pay attention for cluster labels.

The second type of clustering algorithm is Description aware algorithms which give a greater weight age to one specific feature of the clustering process among the rest. Suffix Tree Clustering (STC) a type of Description aware algorithm [7] [8] was proposed, which creates labels which can be understood by the users. It is based on the common phrases that appear in the documents. Thus, STC uses a phrase-based-model for document representation.

Description centric algorithms [9] are designed specifically for WSRC or web document clustering, to balance between the quality of clusters and the labelling. WDC-CSK [1], which is based on the cuckoo search meta-heuristic algorithm, k-means algorithm, Balanced Bayesian Information Criterion, and frequent phrases approach for cluster labelling. These algorithms takes more time to process and retrieve clusters. In this paper, a new description aware algorithm called WSRC-

CSABC has been introduced in order to overcome the headache of local optima and process time.

Cuckoo Search Algorithm

The cuckoo search provides a combined global and local search strategy in the solution space. It is based on the obligate brood parasitic behaviour of some cuckoo species in combination with the Lévy flight behaviour of some birds and fruit flies [10] as shown in Figure 2. It provides a new way of intensification and diversification [11].

Simplifying the breeding behaviour of the cuckoo, a set of three idealized rules can be established [12];

1. Each cuckoo lays one egg at a time, and deposits its egg in a randomly chosen nest;
2. The best nests with high-quality eggs will be carried over to the next generations;
3. The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$ (p_a, Percentage of abandonments). In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

Based on these three rules, the basic steps of cuckoo search can be summarized in Figure 3[1]. In the cuckoo search algorithm, Cuckoo is randomly obtained through Lévy flight using Equation (1).

$$X_i^{t+1} = X_i^t + \alpha + \text{Lévy}(\gamma) \quad (1)$$

Where, $\alpha > 0$ represents a step size, which should be related to the scales of problem the algorithm is trying to solve. In most cases, α can be set to the value of 1, and

$$\text{Lévy}(\gamma) = t^{-\gamma}$$

Where, γ is the step length, $\gamma \in (0, 3]$, and γ is randomly generated using a Lévy distribution.

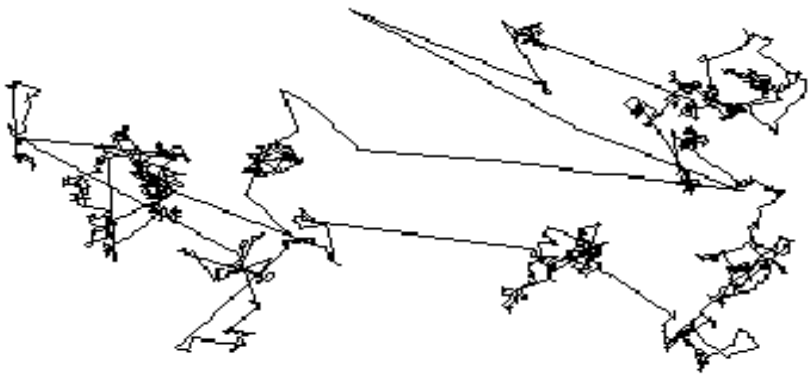

Figure 2: Lévy Flight Distribution

```
01 Objective function f(x); x=(x1, x2, ...., xd)
02 Generate initial population of n host nests xi where i=1, 2, ...., n
03 While stop criterion
04 Get cuckoo randomly by levy flights
05 Evaluate its Quality/Fitness Fi
06 Choose a nest among n randomly
07 if (Fi>Fj)
08 Replace j by new solution
09 A fraction of worst nests are abandoned and new ones are built
10 Keep the best solutions
11 Rank the solutions and find the current best.
```

Figure 3: Algorithm of Cuckoo Search Algorithm via Lévy Flight [1]

In 2013 [17], a comparative study of Genetic Algorithm, Particle Swarm Optimization, and Cuckoo Search over several clustering problems was conducted. CS shows better results in average classification error percentage and in execution time. In this work, a pre-defined value of clusters should be used in order to execute all algorithms; therefore, this proposal is unfeasible for web document clustering. Further, it does not pay attention to cluster labels, noise, and other success factors of the specific research field.

Proposed Work

In this section, a detailed description of Artificial Bee Colony Algorithm is discussed.

Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm which is one of the most recently introduced optimization algorithms simulates the intelligent foraging behaviour of a honey bee swarm. ABC algorithm can efficiently be used for multivariate data clustering. Artificial Bee Colony (ABC) algorithm was proposed by Karaboga for optimizing numerical problems. It is a very simple, robust and population based stochastic optimization algorithm. The performance of the ABC algorithm is compared with those of other well-known modern heuristic algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO) on constrained and unconstrained problems.

In ABC algorithm, the colony of artificial bees contains three groups of bees: onlookers, employed bees and scouts. A bee which is waiting on the dance area for making a decision to choose a food source is called onlooker. A second kind of bee going to the food source visited by it before is named employed bee. The third kind of bee is scout bee that carries out random search for discovering new sources.

Step 1: Load the samples
Step 2: Generate initial population
Step 3: Set Cycle to 1
Step 4: For each employed bee
{
Produce new solution v_i
Apply greedy selection procedure
}
Step 5: Calculate probability p_i for each z_i
Step 6: For each onlooker bee
{
Select a solution z_i depending on p_i
Produce new solution v_i

Apply greedy solution process

}
Step 7: If there is an empty solution for scout then substitute it with a new solution which is randomly produced
Step 8: Cycle=Cycle+1
Step 9: until Cycle=MCN

Figure 4: Pseudo-code of the Artificial Bee Colony Algorithm

In the algorithm, the colony consists of employed artificial bees and the onlookers, where both of them are in equal number. The number of the employed bees or the onlooker bees is equal to the number of solutions (the cluster centers) in the population. At the initial step, the ABC generates a randomly distributed initial population P(C =0) of SN solutions (food source positions), where SN represents the size of population. Each solution z_i where i=1, 2, ..., SN is a D-dimensional vector, where D is the number of product of input size and cluster size for each data set. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C =1, 2, ..., MCN, of the search processes of the three kinds of bees present i.e. employed, onlooker and scout.

An employed bee produces a change on the places (solution) in her memory depending on the visual information and tests the nectar amount i.e. fitness value of the new source i.e. new solution. Given that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the recent place and forgets

the previous one. Otherwise she keeps the place of the old one in her memory. After all employed bees finish the searching procedure, they share the nectar information i.e. fitness value of the food sources and their place information with the onlooker bees on the dance area.

An onlooker bee measures the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. The employed bee, they make change on the place in her memory and checks the nectar amount of the candidate source. Providing that its nectar is greater than that of the old one, the bee remembers the recent position and forgets the old one. An artificial onlooker bee selects a food source depending on the probability value associated with that food source, p_i which is calculated by the following expression (2):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{2}$$

Where, SN is the number of food sources equal to the number of employed bees, and fit_i is the fitness of the problem.

In order to produce a candidate food position from the old one in memory, the ABC uses the following expression (3):

$$v_{ij} = z_{ij} + \varphi_{ij}(z_{ij} - z_{kj}) \tag{3}$$

Where, k ∈{1,2,...,SN} and j∈{1,2,...,D} are randomly chosen indexes. Although k is found randomly, it has to be different from i. φ_{ij} is a random number between [−1, 1]. It controls the production of neighbour food sources around z_{ij} and represents the comparison of two food positions visible to a bee. We can infer from (6), as the difference between the parameters z_{ij} and z_{kj} decreases, the perturbation on the position z_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is reduced. The food source of which the nectar is not visited by the bees is substituted with a recent food source by the scouts. In ABC, this is simulated by generating a position randomly and replacing it with the empty one. In ABC, providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be left out. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "limit" for abandonment. Assume that the left out source isz_i and j∈ {1,2,...,D}, then the scout discovers a recent food source to be replaced with z_i.This operation can be defined as in (4)

$$z_i^j = z_{min}^j + \text{rand}(0,1)(z_{max}^j - z_{min}^j) \tag{4}$$

After each candidate source position v_{ij} is produced and then evaluated by the artificial bee, its functioning is compared with that of its old one. If the recent food source has equal or better nectar than the previous source, it is replaced with the previous one in the memory. Otherwise, the previous one is retained in the memory. In other way, a greedy selection process is employed as the selection operation between the previous and the candidate one.

In this paper Artificial Bee Colony algorithm is used to cluster the web search results. Cuckoo Search algorithm calculates a quality for each and every snippet fed

into it. Based on this quality ABC algorithm performs a global search and clusters the documents of same quality so that the result obtained is more efficient when compared to other algorithms that is used for clustering.

Experimentation

Data Sets For Validation

The proposed new algorithm, i.e. WSRC-CSABC was used for clustering of web results on four traditional benchmarking data sets such as DMOZ-50, AMBIENT, MORESQUE and ODP-239.

Dmoz-50 data set consists of 50 queries derived from Open Directory Project. Each query has on average 129.14 documents, 6.02 subtopics and 22.62 relevant results per retrieved subtopic.

This dataset does not contain query keywords instead each query is a collection of documents.

The Dmoz-50 is freely downloadable.

Ambient (AMB Iguous ENTries) data set consists of 44 queries extracted from ambiguous entries. Each query has approximately 50.55 ranked search results collected from Yahoo!, 7.91 subtopics, and 7.72 relevant results per retrieved subtopic. The words available in this data set are single worded.

Moresque (MORE Sense-tagged QUEry results) data set consists of 114 ambiguous queries which were conducted as a complement to AMBIENT data set. In this data set queries of different lengths are present, ranging from 1 to 4 words. MORESQUE data set provides 114 queries of length 2, 3 and 4, together with an average of 53.54 top results from Yahoo!, 3.82 subtopics, and 19.43 relevant results per retrieved subtopic.

Odp-239 data set consists of 239 queries derived from Open Directory Project (http://www.dmoz.org). Each query has on average 106.95 documents, 9.56 subtopics, and 11.38 relevant results per retrieved subtopic. ODP-239 consists of many small collections, each with a comparatively large set of classes, as opposed to having one large collection of documents with a small number of classes. The topics, subtopics, and their associated documents were selected in such a way that the distribution of documents across subtopics reflects the relative importance of subtopics.

Existing Systems

WSRC-CSABC was compared with WDC-CSK, STC and Lingo from two perspectives, the quality of the clustering results (based on precision, recall, F-measure and accuracy) and the ease in which users can use clustering results (based on SSL_k measure). STC [15] is the original web search clustering approach based on suffix trees and frequent phrases and Lingo [16], which is a well-known successor of STC. In this web clustering algorithm (implemented in the Carrot2open source framework) frequent phrases of documents are extracted using suffix arrays, then the best frequent phrases are extracted using Singular Value Decomposition (SVD), and finally documents are allocated to such frequent phrases.

Ground-Truth Validation

It aims at assessing how good a clustering method is at recovering known clusters from a standard partition. Several evaluation parameters namely precision, recall, F-measure and Accuracy are available. Table 1 shows results of each measure for the AMBIENT data set and algorithm.

Table 1. Results of Each Measure For AMBIENT Dataset

ALGORITHM	PRECISION	RECALL	F-MEASURE	ACCURACY	SSL_k
WDC-CSK	73.71	59.06	62.02	81.5	111.66
WSRC-CSABC	89.80	55.70	68.75	80.2	161.22
STC	86.75	50.4	58.68	80.43	160.46
LINGO	72.40	53.14	55.38	81.89	187.44

On the AMBIENT data set, WSRC-CSABC outperforms WDC-CSK in precision, F-measure and SSL_k and it outperforms LINGO and STC in precision, recall, F-measure. WSRC-CSABC gives more promising results when compared to other algorithms. WSRC-CSABC gives more efficient result when compared to WDC-CSK. The time complexity of the proposed algorithm is very less when compared to WDC-CSK. In Figure.5, the curves of precision, recall, F-measure through different number of snippets fed as an input are shown.

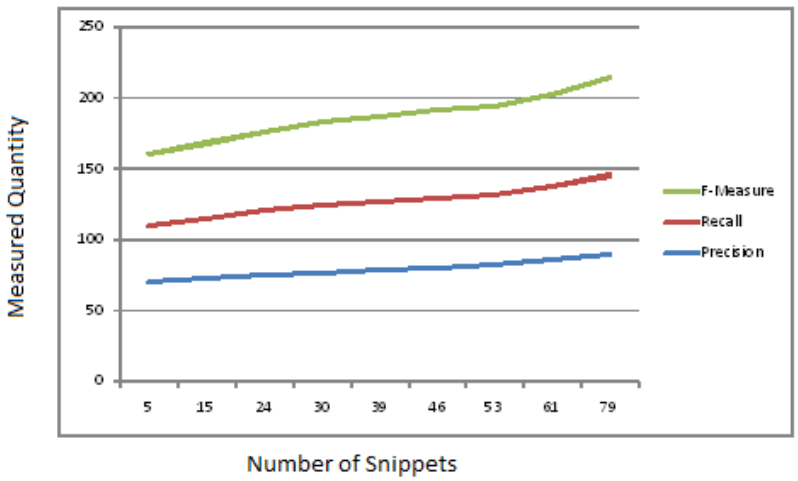

Figure 5: Observation of Parameters

Table 2: Time taken for the retrieval of search results

Number of Snippets	WSRC-CSABC	WDC-CSK
5	0.022	3.123
15	0.013	4.395
24	0.027	2.068
30	0.021	2.64
39	0.014	1.851
46	0.019	1.342
53	0.041	3.118
61	0.054	2.592
79	0.134	2.788

From the above table (2), we conclude that WSRC-CSABC retrieves the results more efficiently when compared to WDC-CSK and the time of retrieval of the search results is also very less.

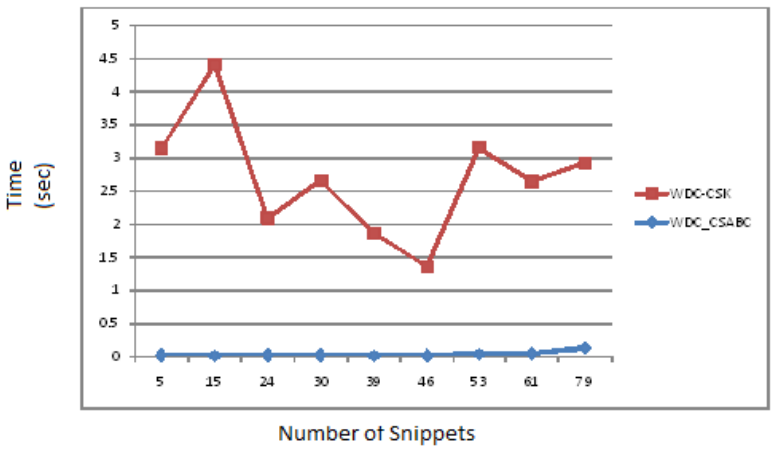

Figure 6: Retrieval time comparison of WSRC-CSABC and WDC-CSK

Conclusions and Future Work

The WSRC-CSABC algorithm has been successfully designed, implemented and evaluated. WSRC-CSABC is a description-centric algorithm for the clustering of web results based on the cuckoo search algorithm and the artificial bee colony algorithm with the capacity of automatically defining the number of clusters. WSRC-CSABC shows excellent experimental results on traditional benchmark data sets of the research area. There are several tasks for future work, among them: (i) Use of WordNet or other semantic tool to work with concepts instead of terms and comparing the results with other state of the art algorithms, e.g. using concepts related with proximity based collaborative clustering and (ii) Use of disambiguation

techniques in order to improve quality of cluster results and compare results with other algorithms.

References

[1] Carlos Cobos, Henry Muñoz-Collazos, Richar Urbano-Muñoz, Martha Mendoza, Elizabeth León , Enrique Herrera-Viedma, Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion, Information Sciences 281 (2014) 248–264.

[2] C. Carpineto, S. Osin´ ski, G. Romano, D. Weiss, A survey of Web clustering engines, ACM Comput. Surv. 41 (2009) 1–38.

[3] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (2002) 443–462.

[4] R. Baeza-Yates, A.B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., 1999.

[5] P. Berkhin, J. Kogan, C. Nicholas, M. Teboulle, A survey of clustering data mining techniques, in: S. Sri (Ed.), Grouping Multidimensional Data, Springer- Verlag, 2006, pp. 25–71.

[6] K. Hammouda, Web Mining: Clustering Web Documents A Preliminary Review, 2001, pp. 1–13.

[7] Y. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequences, Data Knowl. Eng. 64 (2008) 381–404.

[8] Z. Oren, E. Oren, Web document clustering: a feasibility demonstration, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Melbourne, Australia, 1998, pp. 46–54.

[9] F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, in: KDD ‘02: International conference on Knowledge discovery and data mining (ACM SIGKDD), ACM, Edmonton, Alberta, Canada, 2002, pp. 436–442.

[10] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms (2008) 128.

[11] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[12] Y. Xin-She, S. Deb, Cuckoo search via lévy flights, in: World Congress on Nature & Biologically Inspired Computing, 2009.NaBIC 2009, 2009, pp210– 214.

[13] M.H. Chehreghani, H. Abolhassani, M.H. Chehreghani, Density link-based methods for clustering web pages, Decis. Support Syst. 47 (2009) 374–382.

[14] M. Hemalatha, D. Sathyasrinivas, Hybrid neural network model for web document clustering, in: Second International Conference on the Applications of Digital Information and Web Technologies, 2009. ICADIWT ‘09, 2009, pp. 531–538.

[15] Z. Oren, E. Oren, Web document clustering: a feasibility demonstration, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Melbourne, Australia, 1998, pp. 46–54.

[16] S. Osin´ ski, D. Weiss, A concept-driven algorithm for clustering search results, Intell.Syst. 20 (2005) 48–54.

[17] J. Senthilnath, V. Das, S.N. Omkar, V. Mani, Clustering using Lévy flight cuckoo search, in: J.C. Bansal, P. Singh, K. Deep, M. Pant, A. Nagar (Eds.), Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Springer, India, 2013, pp. 65– 75.