International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 6 (2017) pp. 995-1008

Text Clustering Quality Improvement using a hybrid Social spider optimization

T. Ravi Chandran, A.V.Reddy,B.Janet

Dept. of Computer Applications, NIT, Trichy, India.

ABSTRACT

Text document clustering is one of the most widely studied data mining problems. It organizes text documents into groups such that each group has similar text documents. While grouping text documents, several issues have been observed. Accuracy and Efficiency are the main issues in text document clustering. Recently, as clustering problem can be mapped to optimization problem, evolutionary optimization techniques have been used by researchers to improve accuracy and efficiency. Evolutionary techniques are stochastic general purpose methods for solving optimization problems. Swarm Intelligence is one such technique that deals with aggregative behavior of swarms and their complex interactions without any supervision.

In this paper, we proposed a novel swarm intelligence algorithm called Social Spider Optimization SSO for textual document clustering. We compared it with K-means clustering and other state-of-art clustering algorithms such as PSO, ACO and Improved Bee Colony Optimization and found it to give better accuracy. Then we proposed two hybrid clustering techniques namely SSO + K-means and K-means + SSO and found SSO + K-means clustering outperformed K-means + SSO, KPSO(K-means + PSO) , KGA (K-means + GA), KABC(K-means + Artificial Bee colony) and Interleaved K-means + IBCO clustering techniques. We used Sum of intra cluster distances, average cosine similarity , accuracy and Inter cluster distance to measure the performance of clustering techniques.

KEYWORDS: Text document clustering, Evolutionary optimization techniques, Swarm intelligence, Social spider optimization, K-means clustering

INTRODUCTION

Data clustering is one of the most widely used task in data mining due to its capability for summarizing large data collections [1]. The main aim of a good clustering approach is to minimize intra-cluster distances between data elements and maximize inter-cluster distances [2][3][4]. Data clustering is carried out in a supervised or unsupervised way using two main clustering methods namely partitioning clustering and hierarchical clustering [5].Partitioning clustering techniques are capable of clustering large datasets, because of this, they are most popular in various researchfields [51].Partitioning methods relocate data items by moving them from one cluster to another cluster, starting from an initial partitioning [52]. These methods generally require that the number of clusters will be preset byusers. Other reasons for popularity of these methods include itslinear time complexity, ease of interpretation, simplicity of implementation, speed of convergence and adaptability to work on sparse data [6].

Unlike partitioning clustering, hierarchical clustering produces clusters in the form of a nested tree. There are two different hierarchical clustering approaches namely agglomerative and divisive. The agglomerative method follows a bottom-up approach such that initially each data point is considered as a cluster of its own and at each step merging operation is performed on the two most similar clusters until a single cluster is produced [53]. The divisive method follows a top-down approach such that initially all the data points are assigned to a single cluster and at each step a cluster is selected and is divided into two clusters until no more division is possible [54]. Four traditional agglomerative methods are Single linkage, Complete linkage, Average linkage and Ward′s methods [7].

Hierarchical clustering generates multiple nested partitions, enabling different users to select different partitions according to desired similarity level [55].Both Partitioning and Hierarchical clustering methods have advantages and disadvantages.

Advantages and disadvantages of Partitioning Clustering are as follows [56][6].

Advantages

(i) Relatively scalable and simple.
(ii) Suitable for well-separated datasets
(iii) Linear time complexity
(iv) Speed of Convergence.

Disadvantages

(i) Concept of distance between points in high dimensional spaces is ill-defined
(ii) Poor cluster descriptors
(iii) The number of clusters should be preset by users
(iv) Quality of clustering results is influenced by initialization, noise and outliers
(v) Local optima entrapments occur very frequently
(vi) Non-convex clusters of varying size and density can not be handled

Advantages and disadvantages of Hierarchical Clustering are as follows [56].

Advantages

i) Embedded flexibility in the level of granularity.
ii) Suitable for problems involving point linkages, e.g. taxonomy trees.

Disadvantages

(i) Once the splitting/merging decision is made, corrections can not be made
(ii) Lack of interpretability regarding the cluster descriptors.
(iii) Vagueness of termination criterion.
(iv) Prohibitively expensive for high dimensional and massive datasets.
(v) Severe effectiveness degradation in high dimensional spaces due to the curse of dimensionality phenomenon
(vi) Non linear time complexity
(vii) No back tracking capability

Over the past few years, a number of issues related to number of inputs, procedure of clustering and qualitative output have been observed. The clustering techniques mainly have to deal with efficiency of execution time and accuracy of clusters.

Textual document clustering organizes large text document collections into groups of related text documents. It has significant importance in our lives as we prefer paperless environment, and World Wide Web has become a part of our lives for the last two decades.For better text document organization and improved information retrieval, efficient and effective document clustering methods should be used.

In Vector space model of representing a text document, each text document Doc_i in document set D is represented by a vector. Let n be number of distinct recognizable terms in D and $w_{1,i}$, $w_{2,i}$, $w_{3,i}$,…, $w_{n,i}$ be term weights of n terms in Doc_i. Then Doc_i= ($w_{1,i}$, $w_{2,i}$, ..., $w_{n,i}$) [8]. For calculating these term weights, we used Term Frequency-Inverse Document Frequency(TF-IDF) approach. Though TF-IDF is a relatively old weighing scheme, it is simple and effective [9].TF-IDF weight of a term t_j in a document Doc_i of document set D can be computed using the equation (1). Let |D| be total number of text documents in document set D, and N represents number of text documents in which term t_j ispresent.

$$\text{tf-idf}(t_j, Doc_i, D) = \text{tf}(t, Doc_i) * \text{idf}(t_j, D) \quad (1)$$

where $\text{tf}(t_j, Doc_i)$ is number of times that term t_j present in document Doc_i

and $\text{idf}(t_j, D)$ is $\log(|D|) \div N)$.

Evolutionary techniques are stochastic general purpose methods for solving optimization problems [6]. As clustering problem can be mapped to optimization problem, evolutionary techniques can also be applied. The basic idea in evolutionary techniques is with the help of evolutionary operators and a population of clustering structures, convergence into a globally optimal clustering can be attained [57]. Evolutionary technique mainly uses Selection, element-wise average for intermediate recombination, and mutation as the generic operators [58]. A fitness function is associated with each individual candidate solution, which quantifies the individual's ability to survive and thrive [59]. Genetic algorithms is the most frequently used evolutionary technique in clustering problems [57].

Recently, several optimization techniques have been proposed which influence the efficiency and accuracy of clustering techniques. These optimization techniques deal with clustering issues such as choosing the initial parameters, centroids optimization, convergence to a solution, and local optima trapping. These techniques have already been found to be successful in solving problems such as global optimization, multi-objective optimization and avoiding being trapped in local optima [10][11][12][13]. We can use an optimization technique as a data clustering algorithm or add optimization to the existing data clustering approaches. In optimization based clustering,inter-cluster distances and intra-cluster distances are used as measures for an objective function to be optimized. Overall, optimization based clustering techniques have been found to be very successful in improving accuracy and efficiency of document clustering due to their capability of exploiting and exploring solution search area to produce optimal solution. Swarm Intelligence (SI) is one such optimization technique where different variants of SI have been proposed to either perform clustering independently or add to the existing clustering techniques. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are the two main SI based techniques which have been modeled and tested on different clustering problems so far [5].

BACKGROUND

Swarm intelligence is the collective behavior of decentralized, self-organized systems, natural or artificial[14]. These systems consists of a population of agents that interact locally with one another and with their environment. The inspiration comes from nature, especially biological systems. Even though there is no centralized control, agents in the population follow simple rules.The research areas that have benefited from the simplicity, flexibility and extendibility of SI techniques include traditional optimization problems, multi objective optimization, planning, routing, scheduling, and load balancing [15][16]. Recently a huge growth has been observed in the literature of SI based optimization for Knowledge Discovery Data mining including association rule mining, classification rule mining, sequential pattern mining, data clustering and outlier detection [5].

The social spider optimization (SSO) algorithm is a population based algorithm proposed by Cuevas in 2013[17]. There are two fundamental elements of a social spider colony. They are social members and communal web. The social members are divided into males and females[18]. Spiders of gender female attract or dislike other spiders. Male spiders are classified into two classes, dominant and non-dominant male spiders. Dominant male spiders have better fitness than

non-dominant male spiders.Mating operation allows the information exchange among dominant males and females. A dominant male mates with one or all females within a specific range to produce offspring.Each spider is represented by a position, a weight and vibrations perceived from other spiders. Spider position can be regarded as a candidate solution within the solution search space.Any spider whose fitness is greater than that of spider s_i is considered as a better spider than spider s_i. Any spider whose fitness is the largest of fitness' of all spiders is considered as globally best spider. Any spider whose fitness is the smallest of fitness' of all spiders is considered as the worst spider.

Every spider has a weight based on the fitness value of the solution given by it. The communal web is responsible for transmitting information among spiders. This information is encoded as small vibrations. These vibrations are very important for the collective coordination of all spiders in the solution search space. The vibrations depend on the weight and distance of the spider which has generated them [18].

If total population consists of N spiders, the number of females $N_f$is randomly selected within the range of 65–90% of N and the remaining spiders are considered as male spiders. Each spider position is randomly selected based on upper bound and lower bound of each dimension of objective function as shown in equation (2)

$$s_{i,j} = p_j^{low} + random(0, 1) * (p_j^{high} - p_j^{low}) \quad (2)$$

where p_j^{high} and p_j^{low} are upper bound and lower bound of j^{th} dimension of objective function to be optimized,

$s_{i,j}$ is initial position of spider s_i in j^{th} dimension.

The weight w_i of each spider s_i represents quality of solution given by it. It can be calculated using equation (3).

$$w_i = \frac{fit_i - fit_{worst}}{fit_{best} - fit_{worst}} \quad (3)$$

where $fit_i$is fitness of current spider s_i,

fit_{best} is the smallest of fitness' of all spiders, and

fit_{worst} is the largest of fitness' of all spiders (for minimization problem).

The vibrations perceived as $v_{i,j}$ by spider s_i from spider s_j can be calculated using equation (4).

$$v_{i,j} = w_j * e^{-d^2} \quad (4)$$

where dis the distance between spider s_i, and spider s_j,

and w_j is weight of spider s_j.

Each spider s_i will perceive vibrations $vibc_i$, from a nearest better spider, $vibb_i$ from globally best spider, and $vibf_i$ from nearest female spider. The vibrations perceived by spider s_i can be calculated using equations (5), (6), and (7).

$$vibc_i = w_j * e^{-d^2} \quad (5)$$

where w_j is weight of nearest better spider of spider s_i,

and d is distance between both of them.

$$vibb_i = w_j * e^{-d^2} \quad (6)$$

where w_j is weight of globally best spider of entire population,

and d is distance between both of them.

$$vibf_i = w_j * e^{-d^2} \quad (7)$$

where w_j is weight of nearest female spider of spider s_i

and d is distance between both of them.

The female spiders attract or dislike other spiders irrespective of gender. The movement of attraction or repulsion depends on several random phenomena. A uniform random number r is generated within the range [0, 1]. If r is smaller than PF(threshold probability that female spider attracts any other spider), an attraction(+) movement is generated, otherwise, a repulsion (-)movement is produced. If an attraction is generated, the next position of female spider in search space can be calculated as using equation (8). In this paper, we used the terms spider and spider's position interchangeably.

$$f_i = f_i + \alpha\, vibc_i (f_c - f_i) + \beta\, vibb_i(f_b - f_t) + \delta(\gamma - 0.5) \quad (8)$$

If a repulsion movement is produced, the position of female spider can be calculated using equation (9).

$$f_i = f_i - \alpha\, vibc_i(f_c - f_i) - \beta\, vibb_i (f_b - f_t) + \delta(\gamma - 0.5) \quad (9)$$

In equations (8) and (9),

f_b is position of globally best spider,

f_c is position of nearest better spider of female spider f_i, and

α, β, δ and γ are random numbers between 0 and 1.

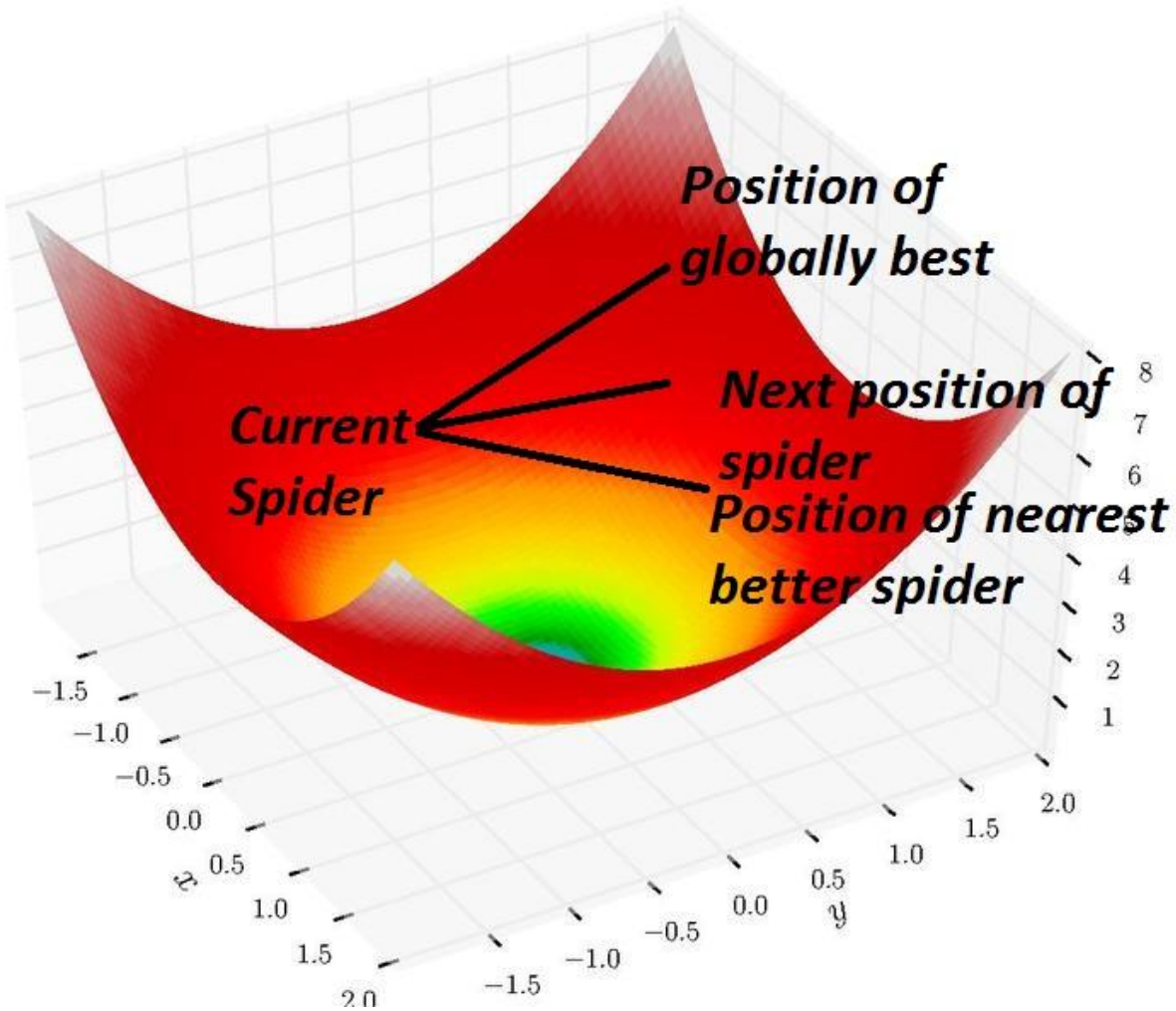

Figure 1. Generation of next position of female spider

A dominant male spider has a weight above the median value of the weights of male population.The other males with weights under the median are called non-dominant males. The next position of dominant male spider can be calculated using equation (10).

$$m_i = m_i + \alpha\ vibf_i\ (f_s - m_i) + \delta(\gamma - 0.5) \quad (10)$$

where f_s is position of nearest female spider of male spider m_i, and

α ,δ and γ are random numbers between 0 and 1.

The position of non dominant male spider m_i can be calculated using equation (11).

$$m_i = m_i + \alpha\ (W - m_i) \quad (11)$$

where W is weighted mean of male spiders.

Weighted mean W of male spiders can be calculated using equation (12).

$$W = \frac{\sum_{h=1}^{N_m} m_h * w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} \quad (12)$$

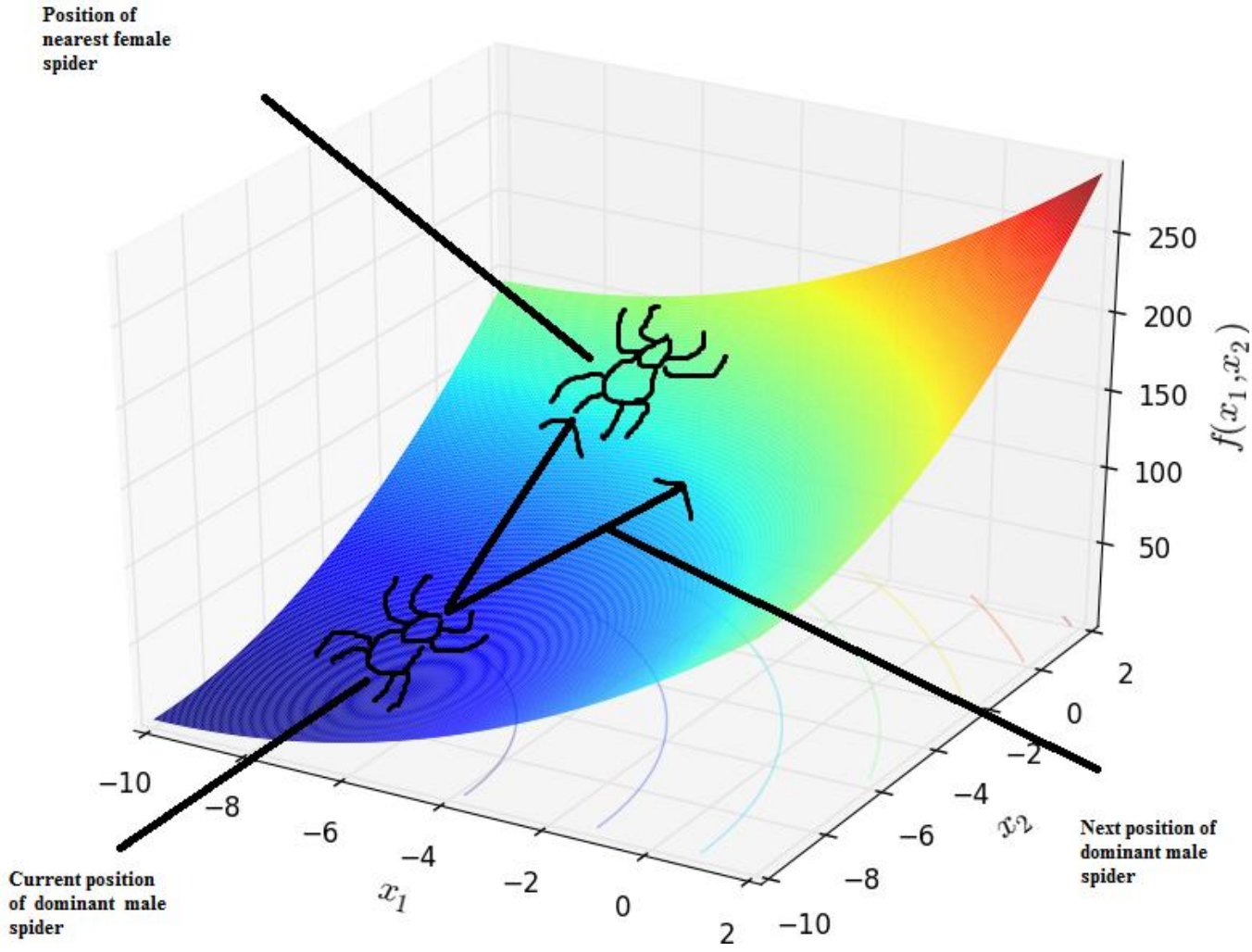

Figure 2. Generation of next position of dominant male spider

Before mating, each dominant male spider has to find a set of female spiders within the specified range of mating.

The range of mating r can be calculated using equation (13).

$$r = \frac{\sum_{j=1}^{n} (p_j^{high} - p_j^{low})}{2 * n} \quad (13)$$

where n is number of dimensions present in objective function,

p_j^{low} and p_j^{high} are lower bound and upper bound of j^{th} dimension of objective function respectively.

The spiders holding a heavier weight are more likely to influence the new spider. The influence probability of each member is assigned by the Roulette Wheal method[18]. From the above equations (8) to (11), it is clear that next position of female spiders is influenced only by positions of globally best spider and nearest better spider. Next position of dominant male spiders is dependent only on position of nearest female spider. Because of this, SSO can search solution space in different directions at the same .

Let M_g be a dominant male spider and E_g be the set of all female spiders within the range of mating operation. Then T_g, the set of all spiders which are participating in mating operation can be calculated using equation (14) as follows.

$$T_g = E_g \cup M_g \quad (14)$$

S_{new}, the position of resultant spider of mating operation can be calculated using Roulette Wheal method using equation (15).

Let k be total number of spiders in T_g.

$$S_{new} = \frac{\sum_{i=1}^{k} (\text{positon of } i^{th} \text{ spider in } T_g * \text{ weight of } i^{th} \text{ spider in } T_g)}{\sum_{i=1}^{k} \text{weight of } i^{th} \text{ spider in } T_g} \quad (15)$$

Algorithm 1gives a description of SSO based problem solving. The selection of objective function depends on the type and nature of the problem. Stopping criteria can be maximum number of iterations or accepted error value.Among N spiders, the first N_f spiders are considered as female spiders and remaining spiders are considered as male spiders.

Algorithm 1. SSO for finding Minima

Input: function f with number of dimensions n, lower bound p^{low}, and upper bound p^{high} of each dimension of f

Output: Minima M (values of each dimension at which value of function f becomes minimum)

Parameters: Swarm size S, Threshold PF, and Maximum number of iterations Max

1. Set the initial value of total number of solutions N in the population size S, threshold PF,and maximum number of iterations Max

2. Compute N_f number of female spiders, N_m number of male spiders using the following formulae.

N_f= floor[(0.9-random(0,1)*0.25)* N]

$N_m = N - N_f$

3. Initialize current iteration, Iteration with 0

Iteration = 0

4. Initialize randomly each female spider as follows

for(k = 1; k < N_f+ 1; k + +) do

for(j = 1; j < n + 1; j + +) do

$f_{i,j} = p_j^{low} + random(0, 1) * (p_j^{high} - p_j^{low})$

end for

end for

5. Initialize randomly each male spider as follows.

for(k = 1; k < N_m+ 1; k + +) do

for(j = 1; j < n + 1; j + +) do

$m_{i,j} = p_j^{low} + random(0, 1) * (p_j^{high} - p_j^{low})$

end for

end for

6. Repeat

a) Calculate weight of each spider s_i using equation(3)

b) Move female spiders according to female cooperative operator using equations (8) and (9)

c) Move male spiders according to male cooperative operator using equations (10) and (11)

d) Perform mating operation within range of matingand replace worst spider with new spider, if weight of new spider > weight of worst spider

e) Iteration ++

until Iteration<= Max

7. Return spider with best fitness as Minima M.

In the above algorithm, each spider represents a solution in search area. That is, each spider s_i is a collection of values of each dimension. If we substitute those values in the given function, function value will be obtained. The function value can be taken as fitness of spider s_i. And before mating operation, Each dominant male spider identifies all female spiders whose function values are less than or equal to r, range of mating operation.

RELATED WORK

We will now outline some of the related work that has tackled different issues of data clustering in recent years.Yang Yang, et al [19] presented a novel clustering model, namely Multi Task Spectral Clustering (MTSC) to handle correlations among related clustering tasks, the problem of clustering out-of-sample data and the discriminative property of cluster label matrix. The authors clearly revealed that clustering performanceis significantly influenced by exploring of the inter task correlations and integration of discriminative information. But time complexity of MTSC is more than CLGR (Clustering with local and global regularization).Ying he, et al [20] used ensemble learning and proposed a new clustering algorithm to improve the performance of traditional H-K clustering algorithm,

In high dimensional dataset. The authors called the proposed asEPCAHK (Ensemble Principle Component Analysis Hierarchical K-means Clustering algorithm). Liping Jing, et al [22] very recently proposed a stratified sampling method. They used it in ensemble clustering of high dimensional data for generating subspace component data sets. But the diversity of the component clusterings is sacrificed to some extent.Xinquan Chen [23] presented new clustering algorithms namely Clustering based on Near Neighbor Influence (CNNI), an improved version in time cost of CNNI algorithm (ICNNI), and a variation of CNNI algorithm (VCNNI). They are inspired by the idea of near neighbors and the superposition principle of influence. They showed that proposed algorithms are better than some classical clustering algorithms in terms of accuracy and efficiency. But, because of higher computational complexity,

the proposed algorithms can not be applied on big data.Chien-Hsing Chen, et al [25] proposed a new Feature Selection method to identify salient features that are useful for maintaining the instance's nearest neighbors and farthest neighbors. The mutual information criterion is used by proposed algorithm to estimate feature salience by considering

maintainability. The proposed algorithm outperformed the filter-based methods for identifying features that are useful in clustering process. Clustering. But the algorithm could not handled noisy features.Chen Qin,et al [26] proposed a novel unsupervised distance metric learning algorithm. They used the proposed algorithm to maximize a stochastic variant of the leave-one-out K-nearest neighbor (KNN) score on unlabeled data. The authors demonstrated that proposed algorithm can be effectively applied to tasks of joint dimensionality reduction and clustering. Guojun Gan, et al [27] proposed a subspace clustering algorithm for clustering high dimensional data. The proposed algorithm has the capability of automatic feature grouping . In addition, a new component is introduced into the objective function to capture the feature groups and a new iterative process is defined to optimize the objective function so that the features of high-dimensional data are grouped automatically. But Initialization of cluster centroids has an influence on clustering results..

We will now outline some of the related work that has tackled different issues of text document clustering in recent years.Hrishikesh Bhaumik, et al [29] proposed a new approach for clustering English text documents, based on finding the pair wise correlation of documents in a given set of text documents. The correlation coefficient for each pair of documents is calculated on the basis of ranks given to the words in the documents. The ranking of the words occurring in a document is computed on the basis of weights of the words calculated according to the conventional TF-IDF factor. Proposed method of text classification using correlation coefficient can successfully classify a set of given text documents, when the total number of classes present in the data set is not known a priori. But, Complexity of the proposed algorithm was not optimized. Ximing Li [31] proposed an Adaptive Centroid based Clustering algorithm for text document data that begins with hundreds of small clusters for acceptable CFC vectors, and then iteratively regroups clusters of documents until convergence is achieved. ACC achieves competitive performance with the state-of-art clustering approaches on both balanced and unbalanced datasets.The main drawback of ACC is its slow convergence.Nilupulee Nathawitharana, et al [32] proposed new methodology which considers the feature overlap between the clusters when clustering text documents. Hierarchical clustering facilitated by the Growing Self-Organizing Map (GSOM) is used together with the calculated feature overlap to check the possibility of obtaining clusters with minimum feature overlap. The main deficiency of their study is that experiments using a collection of close categories were not conducted. Yang Yan,et al,[33] carried out their work by incorporating some prior knowledge in the form of pair-wise constraints provided by users into the fuzzy co-clustering framework. Each constraint specifies whether a pair of documents “must” or “cannot” be clustered together. The efficiency of proposedalgorithm is higher than the other semi-supervised clustering approaches. The proposed algorithm ignores constraints on word domain. Sunghae Jun, et al, [34] proposed a new method to overcome the sparsity problem of document clustering. They used a combined clustering method using dimension reduction and *K*-means clustering based on support vector clustering. The proposed approach efficiently manages intellectual property. To define each cluster, only the top ranked terms were used, but experts’ knowledge could be added to define the clusters more effectively.Malik Tahir Hassan [36] proposed an algorithmic framework for partitional clustering of documents that maximizes the sum of the discrimination information provided by documents. It exploits the semantic that term discrimination information provides to yield clusters that are describable by their highly discriminating terms.

We will now outline some of the related work that has tackled different issues of text document clustering using Swarm intelligence in recent years. Rana Forsati [37] proposed an improved bee colony optimization algorithm withan application to document clustering. She introduced cloning, fairness concepts into BCO to make it more efficient for text document clustering. To overcome the shortage of BCO algorithm in searching locally, she hybridize it with the *k*-means algorithm to take advantage of fine tuning power of the widely used *k*-means algorithm which demonstrates good result in local search. The results showed that proposed algorithms are robust enough to be used in many applications compared to k-means and other recently proposed evolutionary based clustering algorithms. The proposed algorithm does not work when the number of clusters is not known or the data points are dynamically added or removed. Kusum Kumari Bharti, et al, [38] used chaotic map as a local search paradigm to improve exploitation capability of Artificial bee colony optimization. The experimental evaluation revealed very encouraging results in terms of the quality of solution and convergence speed. Leticia Cagnina, et al [39] proposed an efficient particle swarm optimization approach to cluster short texts. They extended a discrete PSO algorithm with modifications such as a new representation of particles to reduce their dimensionality, and a more efficient evaluation of the function to be optimized i.e. the Silhouette coefficient. But with larger corpora a constant deterioration in the *F*- measure values was observed, as the number of documents increased. Stuti Karol, et al [40] proposed an evaluation of text document clustering approach based on particle swarm optimization. The proposed approach hybridizes Fuzzy C-means algorithm and K-means algorithm with Particle Swarm Optimization (PSO). The performance of these hybrid algorithms has been evaluated against traditional partitioning techniques. The authors concluded that FCPSO deals better with overlapping nature of dataset. Coming bach to SSO, nobody has used Social spider optimization to cluster text documents.

We will now outline some of the research work carried out on Social spider optimization technique. Erik Cuevas, et al [47] proposed a swarm optimization algorithm for solving optimization tasks. The algorithm is based on the simulation of the cooperative behavior of social-spiders. Proposed SSO delivered better results than PSO and ABC for all bench mark functions. And the authors observed that rate of convergence of SSO is the fastest, which found the best solution in less of 400 iterations on average. The same authors Erik Cuevas, et al [48]proposed SSO for solving constrained optimization tasks. They named the new algorithm SSO-C that incorporates the combination of two different paradigms namely a penality

function, and a feasibility criterion. James J.Q. Yu, et al [46] proposed a new meta heuristic for global optimization and called it as Social Spider Algorithm (SSA). They showed that the performance of SSA in solving multimodal optimization problems is superior. Priyadharshini, V., et al [45] suggested a SSO for optimizing the web service during publishing operation. This approach brings the outcome of high performance for searching a service with many benchmark functions. The authors considered a very few negative attributes into account for evaluating services. Pereira, D.R., et al [43] proposed a social-spider optimization approach for Support vector machines parameters tuning . They used SSO for finding suitable values for SVM parameters. The authors showed that SSO is a suitable tool to perform a global search to find out suitable hyper parameters that maximize the recognition rate. But proposed SSO is slower than Novel Global Harmony Search(NGHS). Later the same authors Pereira, D.R., et al [44] used SSO for improving training phase of Artificial Neural Networks and validated proposed approach in the context of Parkinson's disease recognition. And they found that proposed approach is not as fast as Self-adaptive Global Harmony Search (SGHS).

SOCIAL SPIDER OPTIMIZATION (SSO) BASED TEXT DOCUMENT CLUSTERING

In SSO based text document clustering, each spider represents a collection of clusters. The algorithm starts with initializing each spider with K randomly chosen text documents where K is number of clusters to be formed. These K text documents in each spider s_r will be treated as K initial centroids. Each text document in the dataset is associated with exactly one of these K centroids based on distance measure. Then we calculate fitness and weight of each spider using equation (12) and equation (3) respectively. The fitness of each spider s_r is average distance between text documents and cluster centroid.. Assume that clusters to be formed are $C_1, C_2, C_2, \ldots, C_K$, Then fitness fit_r of spider s_r can be calculated using equation (12)

$$fit_r = \frac{\sum_{i=1}^{K} \frac{\sum_{j=1}^{n_i} \text{distance}(\text{centroid}_i, \text{doc}_j)}{n_i}}{K} \quad (12)$$

where

$centroid_i$ is centroid of cluster C_i,

doc_j is j^{th} text document present in cluster C_i,

n_i is number of text documents in cluster C_i,

K is number of clusters in each spider, and

distance is distance measure function that takes two document vectors.

Smaller the average distance between documents and the cluster centroid, the more compact the clustering solution is [41]. Hence we can consider text document clustering problem as minimization problem.

Each spider position is changed according to its cooperative operator. Mating operation is performed on each dominant male spider and a set of female spiders within the range of mating. This process is repeated until stoping criteria is met. SSO based text document clustering is summarized in algorithm 2.

Algorithm 2. SSO based text document clustering

Input: dataset

Output: clusters of relevant text documents

Parameters: Swarm size S, threshold PF, number of clusters to be formed K and Stopping criteria (eg. maximum number of iterations Max)

1. Set the initial value of total number of solutions N in the population size S, threshold PF,and maximum number of iterations Max

2. Compute N_f number of female spiders, N_m number of male spiders using the following formulae.

N_f= floor[(0.9-random(0,1)*0.25)* N]

N_m= N – N_f

3. Initialize randomly each spider with K randomly chosen text documents that can be considered as K initial cluster centroids.

4. Repeat

a) Assign each text document to nearest cluster centroid

b) Calculate fitness of each spider using equation (12)

c) Calculate weight of each spider using equation (4)

d) Move female spiders according to female

cooperative operator using equations (8) and (9)

e) Move male spiders according to male

cooperative operator using equations (10) and (11)

f) Perform mating operation within the specified range of matingand replace worst spider with new spider

if weight of new spider > weight of worst spider

Until stop criteria is met

5.Return spider with best fitness

Generation of random numbers α, β, δ and γ plays an important role in SSO. The beta probability distribution preserves diversity, avoids the premature convergence and helps to explore hidden areas in the search space during the optimization process. So, in the above proposed clustering algorithms SSO and Modified SSO, beta distribution can be used to generate random numbers so that better clustering results can be achieved.

Hybridized SSO Based Text Document Clustering :

SSO clustering algorithms produce more compact clustering than K-means clustering because of their globalized searching ability.But K-means is efficient for large datasets in terms of execution time. SSO clustering algorithms converge only after 100 iterations whereas K-means converges within 20 to 25 iterations on datasets shown in Table 1. The K-means clustering is summarized in algorithm 3.

Algorithm 3. K-means text document clustering

1. Initialize K cluster centroid vectors with randomly chosen K text documents from dataset
2. Assign each text document in dataset to nearest cluster centroid
3. Recalculate centroid of each cluster using equation (13)

$$centroid_i = \frac{\sum_{j=1}^{n_i}, doc_j}{n_i} \quad (13)$$

where doc_j represents j^{th} text document vector that belongs to cluster $_i$ and

n_i represents the number of text documents present in cluster $_{i.}$

4. Repeat steps 2 and 3 until stop criteria is met

In hybridized SSO based text document clustering, the ability of globalized searching of SSO algorithm and the fast convergence of K-means algorithm are combined in order to produce optimal clustering solution. SSO + K-means algorithm includes two modules namely SSO module and K-means module. At the initial stage, SSO module is used for discovering the vicinity of optimal solution by a global search. The global search of SSO produces K clusters' centroids. These centroids are then passed to K-means module for refining and generating the final optimal clustering solution. The entire process is summarized in algorithm 4.

Algorithm 4. SSO + K-means based text document clustering

1. Start SSO clustering process until maximum number of iterations is exceeded
2. Inherit clustering results from SSO as the initial cluster centroids of K-means clustering process.
3. Start K-means clustering process until maximum number ofiterations is reached.

Hybrid K-means + SSO algorithm includes two modules namely K-means module and SSO module. At initial stage K-means module is executed and it produces K optimal centroids . These K optimal centroids are passed to SSO module to initialize. The entire process is summarized in algorithm 5.

Algorithms 5. K-means + SSO based text document clustering

1. Start K-means clustering process until maximum number of iterations is exceeded
2. Inherit K-centroids from K-means to initialize K spiders
3. Perform Steps 4 and 5 of SSO clustering process.

EXPERIMENTS AND RESULTS

The proposed clustering approaches SSO, SSO + K-means and K-means + SSO are applied on the four datasets collected from PatentCorpus5000 [50]. Each document gives only technical description of the patent, it does not have claims, classes, names of attorneys or inventors. All files are available in ASCII format. No parameter setting is required for K-means. In all SSO based algorithms , we set number of spiders to 50 and threshold probability PF to 0.7.

Table 1: Summary of Text document datasets

	Dataset1	Dataset2	Dataset3	Dataset4
Number of documents	70	108	160	89
Number of terms	15332	20007	90653	18042
Number of clusters	6	7	8	5

We carried out several different experiments as follows.

a. SSO clustering results

Table 2: SSO clustering

Dataset	SICD	Avg. Cosine similarity	F-Measure	Accuracy
Patent dataset 1	5612.04	0.9965	0.9721	97.4621
Patent dataset 2	885.98	0.9938	0.9522	95.0561
Patent dataset 3	120141.45	0.9156	0.9123	92.675
Patent dataset 4	14100	0.9892	0.8478	85.2304

The results of SSO clustering are specified in Table 2. We found that as we increase number of iterations, Sum of intra cluster distances is reduced but accuracy, average cosine similarity and F-measure values are increased. Table 3 specifies how Sum of intra cluster distances is reduced as we increase number of iterations. As we increase number of spiders, more and more worst spiders will be replaced by new better spiders from mating operation, resulting in low SICD.

Table 3: SSO clustering : SICD variation with number of iterations

Dataset	100 iterations	150 iterations	200 iterations	250 iterations	300 iterations
Patent dataset 1	6000.12	5823.56	5800.55	5735.12	5612.04
Patent dataset 2	1204.27	1000.55	991.45	926.45	885.98
Patent dataset 3	130800.89	130003.11	120903.69	120578.09	120141.45
Patent dataset 4	19100.26	18400.69	16252.67	14976.46	14100

We studied the effect of parameter PF(threshold probability) on Sum of intra cluster distances and found that when PF is increased and PF <= 0.8 better results are produced but when PF reaches closer to 1, the results are not better as shown in Fig 3 due to reduced solution space and probability that female spider dislikes another spider reaches closer to zero.

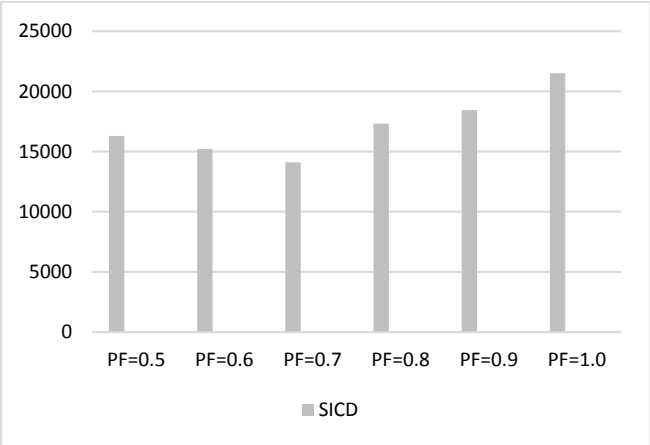

Figure 3. Effect of increase in PF on Sum of intra cluster distances when number of iterations is set to 300 for patent dataset 4

We studied the effect of random variables α, β, δ and γ on Accuracy in SSO based text document clustering, when number of iterations is set to 300. For this, we generated random variables using Beta distribution and found a slight better Accuracy is produced when we used Beta distribution as shown in Fig 4.

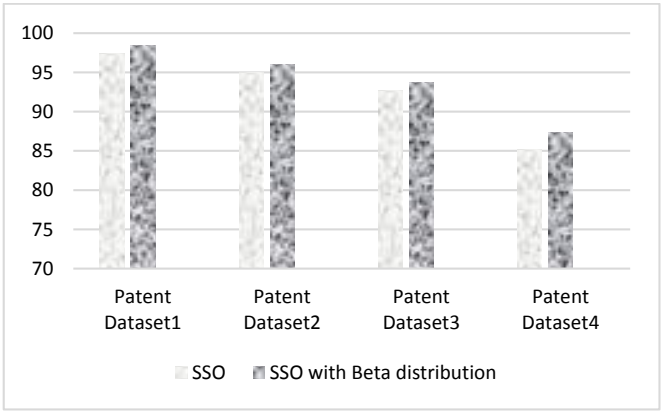

Figure 4. Comparison of Accuracy of SSO against SSO with beta distribution

We found that when Euclidian distance function is used in SSO clustering, better average cosine similarity and accuracy are produced when compared with Manhatton distance function, as Euclidian distance function is not influenced by very small differences in corresponding attribute values unlike Manhatton distance function. That is, the data files that have very small Euclidian distance will more likely be placed in same cluster.

Table 4: SSO clustering : Effect of Distance functions

Dataset	Euclidian distance function		Manhatton distance function	
	Accuracy	Cosine similarity	Accuracy	Cosine similarity
Patent dataset 1	97.4621	0.9965	97.0625	0.9835
Patent dataset 2	95.0561	0.9938	94.6591	0.9898
Patent dataset 3	92.675	0.9156	92.078	0.9009
Patent dataset 4	85.2304	0.9892	84.8822	0.9721

We also checked the convergence of SSO in Patent dataset 4. In Fig.5 Sum of intra cluster distance remains the same after 300 iterations. This implies that convergence is achieved.

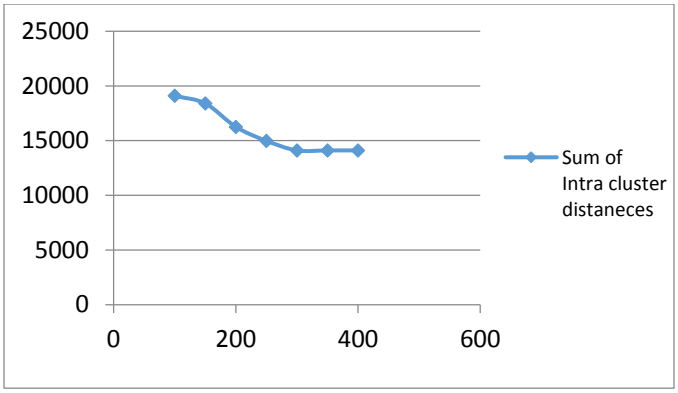

Figure 5. Convergence Analysis for Patent dataset 4 : SSO clustering

Table 5: SSO clustering : Cluster distribution

Dataset	Data per cluster	Sum of Intra cluster distances	Inter cluster distances
Patent dataset 1	10, 12,10,13,10,15	5612.04	580.55
Patent dataset 2	14,17,16,14,15,16,16	885.98	91.45
Patent dataset 3	20,25,17,20,18,23, 22,15	120141.45	1209.69
Patent dataset 4	18,18,18,18,17	14100	162.67

Table 5 shows how clusters are formed when we use SSO clustering. We also measured inter cluster distances when SSO clustering method is used. Intra cluster distance can be defined as sum of square distance between each cluster centroid. Clustering technique should maximize inter cluster distance.

To show the adaptability of SSO clustering for the change in configuration of datasets, we compared results of Random centroid, Random datasets and 10x10 cross validation techinqus on Patent dataset 4. And we found that Cross validation produces more better results as shown in Table 6.

Table 6: Adaptability of SSO clustering

	Random centroids (worst, best)	Random datasets (worst, best)	Cross validation (worst, best)
Sum of Intra cluster distances	14189, 14100	14321, 14255.23	14072, 14067.4
Inter cluster distance	145.08, 162.67	153.55, 159.45	173.14, 174.5692

b. Comparison to other clustering methods

We compared performance of SSO clustering with K-means Clustering, PSO based clustering, ACO based clustering and IBCO clustering and found that SSO clustering produces minimal SICD for all the datasets as shown in Table 7.

Table 7: SICD comparison : Clustering algorithms

Dataset	K-means	PSO	IBCO	ACO	SSO
Patent dataset 1	5723.66	5702.21	5681.56	5692.11	5612.04
Patent dataset 2	905.92	901.08	895.66	889.78	885.98
Patent dataset 3	120614.05	120542.85	120581.43	120807.11	120141.45
Patent dataset 4	14320.44	14400.03	14283.30	14222.61	14100

c. Comparison to hybrid clustering methods

We compared SSO with SSO + K-means, K- means + SSO , KPSO(K- means + PSO), KGA(K- means + GA), KABC(K-means + ABC) and IKIBCO(Interleaved K- means + IBCO) and found SSO+ K-means outperformed other hybrid clustering algorithms as shown in Table 8.

Table 8: SICD comparison : Hybrid Clustering algorithms

Dataset	K-means + SSO	KPSO	KGA	KABC	IKIBCO	SSO + K-means
Patent dataset 1	5720.96	5602.01	5680.88	5692.04	5620.08	5600.04
Patent dataset 2	900.92	899.76	895.08	869.11	873.88	865.98
Patent dataset 3	120504.07	120444	12200.5	12028.35	12090.98	12000.45
Patent dataset 4	14228.89	14200.55	14283.25	14021.55	14091.66	14003.25

CONCLUSION AND FUTURE WORK

So far, no one has applied SSO on document clustering problem. We showed some ways of applying SSO to solve text document clustering problem. We showed how parameters like Threshold probability PF, and random variables effect clustering results. We also showed the effect

of distance measure functions like Euclidian and Manhatton functions on clustering. We used only static structure in implementation. But when number of clusters is not known or text documents are added or removed dynamically, dynamic structure is needed. We think that the dynamic problem is much more challenging and requires careful investigating.

Acknowledgment

We thank Information Processing Lab and Massively Parallel Computing Lab, NIT, Trichy for providing efficient and sufficient resources for our research.

REFERENCES

[1] Malik Tahir Hassan, Asim Karim, Jeong-Bae Kim, Moongu Jeon, "CDIM: Document Clustering by Discrimination Information Maximization", *Information Sciences*, *Volume 316*, *20 September 2015*, *Pages 87-106*

[2] Seema Pralhad Jaybhaye, "Document Clustering for forensic analysis: An approach for improving Computer Inspection", *AJET, Volume 3(70-73), 2015.*

[3] Y. Ioannidis, D. Maier, S. Abiteboul, P. Buneman, S. Davidson,E. Fox, A. Halevy, C. Knoblock, F. Rabitti, H. Schek, andG. Weikum, "Digital library information-technology infrastructures", *Int J Digit Libr, vol. 5, no. 4, pp. 266- 274, 2005.*

[4] P. Cudr´e-Mauroux, S. Agarwal, and K. Aberer, "Gridvine: An infrastructure for peer information management" *IEEE InternetComputing, vol. 11, no. 5, 2007.*

[5] Shafiq Alam, Gillian Dobbie, Saeed Ur Rehman,"Analysis of particle swarm optimization based hierarchical data clustering approaches",*Swarm and Evolutionary Computation*, (*Available online 23 October 2015)*

[6] http://www.ise.bgu.ac.il/faculty/liorr/hbchap15.pdf (accessed on 9th August 2015)

[7] F. Murtagh, P. Contreras, "Algorithms for hierarchical clustering: an overview", WIREs Data Min. Knowl. Discov., 2 (2012), pp. 86–97

[8] https://en.wikipedia.org/wiki/Vector_space_model *(accessed on 5th August 2015).*

[9] Juan Ramos, "Using TF-IDF to determine word relevance in document queries", CiterSeerX.(accessed on 19th August 2015).

[10] C.-Y. Chen, F. Ye, "Particle swarm optimization algorithm and its application to clustering analysis", 2004 IEEE International Conference on Networking, Sensing and Control, vol. 2, IEEE, 2004, pp. 789–794.

[11] S. Alam, G. Dobbie, P. Riddle, "An evolutionary particle swarm optimization algorithm for data clustering", Swarm Intelligence Symposium, 2008. SIS 2008. IEEE, St. Louis, Missouri, USA, 2008, pp. 1–6. http://dx.doi.org/10.1109/SIS.2008.4668294

[12] S. Das, A. Chowdhury, A. Abraham, "A bacterial evolutionary algorithm for automatic data clustering", Proceedings of the Eleventh conference on Congress on Evolutionary Computation, CEC'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2403–2410.

[13] D. Van der Merwe, A. Engelbrecht, "Data clustering using particle swarm optimization", The 2003 Congress on Evolutionary Computation, 2003, CEC'03, vol. 1, IEEE, Canberra, Australia, 2003, pp. 215–220.

[14] https://en.wikipedia.org/wiki/Swarm_intelligence (accessed on 10th August 2015)

[15] D. Merkle, M. Middendorf, H. Schmeck,"Ant colony optimization for resource-constrained project scheduling",IEEE Trans. Evol. Comput., 6 (4) (2002) 333-346.

[16] L.M. Gambardella, É. Taillard, G. Agazzi, Macs-vrptw: "a multiple colony system for vehicle routing problems with time windows", New ideas in Optimization, Citeseer, 1999.

[17] Carlos Eduardo Klein, Emerson Hochsteiner Vasconcelos Segundo, "Modified Social-Spider Optimization Algorithm Applied to Electromagnetic Optimization", *Compumag, 2015.*

[18] Erik cuevas, Miguel Cienfuegos, Daniel Zaldivar, Marco Perezcisneros, " A swarm optimization algorithm inspired in the behavior of social spider", *Expert Systems with applications, vol 40,issue 16, pg 6374-6384, Nov 2013.*

[19] Yang Yang, et al, "Multitask Spectral Clustering by Exploring Intertask Correlation", Cybernetics, Volume 45, issue 5, 1069-1080, 2015

[20] Ying he, et al, " A H-K clustering algorithm based on ensemble learning", Smart and substaninable City, 300-305, 2013

[21] Prasad. K.R, "Assessment of clustering tendency through progressive random sampling and graph-based clustering results", Advance Computing Conference,726-731, 2013

[22] Liping Jing, et al, "Stratified feature sampling method for ensemble clustering of high dimensional data", Pattern Recognition, Volume 48, issue 11,3688-3702, 2015

[23] Xinquan Chen, "A new clustering algorithm based on near neighbor influence", Expert Systems with applications, Volume 42, Issue 21, 7746-7758, 2015

[24] Hong Peng, et al, "An automatic clustering algorithm inspired bymembrane computing", Pattern Recognition Letters, Volume 68, Part 1,34-140, 2015

[25] Chien-Hsing Chen, et al, "Feature selection for clustering using instance-based learning by exploring the nearest and farthest neighbors", Information

Sciences,volume 318,14-27, 2015

[26] Chen Qin,et al, "Unsupervised neighborhood component analysis for clustering", NeuroComputing, Volume 168,609-617,2015

[27] Guojun Gan, et al, "Subspace clustering with automatic feature grouping", Pattern Recognition, Volume 48, Issue 11, 2015

[28] Vispute, S.R, et al, "Automatic text categorization of marathi documents using clustering technique", Advanced Computing Technologies(ICACT), 1-5, 2013

[29] Hrishikesh Bhaumik, et al, "Towards Reliable Clustering of English Text Documents Using Correlation Coefficient", Computational Intelligence and Communication Networks,530-535, 2014

[30] Sarnovsky, M. "Cloud-based clustering of text documents using the GHSOM algorithm on the GridGain platform ",Applied Computational Intelligence and Informatics ,309-313, 2013

[31] Ximing Li, "Adaptive Centroid-based Clustering Algorithm for Text Document Data", Parallel Architectures, Algorithms and Programming (PAAP),63-68, 2014

[32] Nilupulee Nathawitharana, et al, "Feature overlap based dynamic self organizing model for hierarchical text clustering", Industrial and Information Systems,393-398, 2013

[33] Yang Yan,et al, "Fuzzy semi-supervised co-clustering for text documents", Fuzzy Sets and Systems, Volume 215,74–89, 2013

[34] Sunghae Jun, et al, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness", Expert Systems with Applications, Volume 41, Issue 7, 3204-3212, 2014

[35] Rana Forsati,et al, "Efficient stochastic algorithms for document clustering", Information Sciences, Volume 220,269–291, 2013

[36] Malik Tahir Hassan, "Document Clustering by Discrimination Information Maximization", Information Sciences,Volume 316,87-106, 2015

[37] Rana Forsati, "An improved bee colony optimization algorithm withan application to document clustering", Neurocomputing, Volume 159, 9–26, 2015

[38] Kusum Kumari Bharti, et al, "Chaotic gradient Artificial Bee Colony for Text Clustering",Soft Computing (2015)

[39] Leticia Cagnina, et al, "An efficient Particle Swarm Optimization approach to cluster short texts", Information Sciences, volume 265, Pages 36–49, 2014

[40] Stuti Karol, et al, "Evaluation of text document clustering approach based on particle swarm optimization", Central European Journal of Computer Science, Volume 3, issue 2, pages 69-90, 2013

[41] Xin-She Yang, "Recent Advances in Swarm Intelligence and Evolutionary Computation", https://books.google.co.in/books?isbn=331913826X(accessed on 10th August 2015)

[42] Esmin, A.A.A. et al, "Consensus Clustering Based on Particle Swarm Optimization Algorithm", Systems, Man, and Cybernetics,2280-2285, 2013

[43] Pereira, D.R., et al, "A social-spider optimization approach for support vector machines parameters tuning", Swarm Intelligence (SIS), 2014

[44] Pereira, L.A.M, et al, "Social-Spider Optimization-Based Artificial Neural Networks Training and Its Applications for Parkinson's Disease Identification", Computer-Based Medical Systems (CBMS),14-17, 2014

[45] Priyadharshini, V., et al, "A novel Web service publishing model based on social spider optimization technique", Computation of Power, Energy Information and Commuincation,373-387, 2015

[46] James J.Q. Yu, et al, "A social spider algorithm for global optimization", Applied Soft Computing, Volume 30, 614-627, 2015

[47] Erik Cuevas, et al, "A swarm optimization algorithm inspired in the behavior of the social-spider", Expert Systems with Applications,Volume 40, Issue 16, 6374–6384, 2013

[48] Erik Cuevas et al, "A new algorithm inspired in the behavior of the social-spider for constrained optimization", Expert Systems with Applications, Volume 41, Issue 2, 412–425, 2014

[49] James J.Q. Yu,et al, "A social spider algorithm for solving the non-convex economic load dispatch problem", Neuro computing, 2015

[50] www.semanticquery.com/archive/semanticsearchart /downloadsdatacorpus.html (accessed on 4th August 2015).

[51] Shraddha K.Popat, et al, "Review and Comparative Study of Clustering Techniques", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 805-812

[52] Mayank Gupta, et al, "A Performance Evaluation of SMCA Using Similarity Association & Proximity Coefficient Relation For Hierarchical Clustering", International Journal of Engineering Trends and Technology (IJETT) , Volume 15, 2014, 355-359

[53] S.M. Junaid, et al, "Overview of Clustering Techniques", International Journal of Advanced Research inComputer Science and Software Engineering, Volume 4, Issue 11,November2014, 621-624

[54] Johnson J. GadElkarim, et al, "A Framework for

Quantifying Node-Level Community Structure Group Differences in Brain Connectivity Networks", Med Image Comput Comput Assist Interv. 2012; 15(0 2): 196–203.

[55] http://www.ise.bgu.ac.il/faculty/liorr/hbchap15.pdf (accessed on 10th August 2015)

[56] Deepti Sisodia, et al, "Clustering Techniques: A Brief Survey of Different Clustering Algorithms", International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 1 Issue 3 September 2012, 82-87

[57] Oded Z. Maimon, Lior Rokach , "Data Mining and Knowledge Discovery", https://books.google.co.in/books?isbn=0387244352(accessed on 10th August 2015)

[58] Zong Woo Geem, "Music-Inspired Harmony Search Algorithm: Theory and Applications"**,** https://books.google.co.in/books?isbn=3642001858 *(accessed on 10th August 2015)*

[59] http://functionspace.com/articles/64/Soft-Computing-4--Introduction-to-Evolutionary-Algorithms (accessed on 10th August 2015)