Ravi Chandran Thalamala*, A. Venkata Swamy Reddy and B. Janet

A Novel Bio-Inspired Algorithm Based on Social Spiders for Improving Performance and Efficiency of Data Clustering

https://doi.org/10.1515/jisys-2017-0178
Received April 26, 2017; previously published online February 14, 2018.

Abstract: Since the last decade, the collective intelligent behavior of groups of animals, birds or insects have attracted the attention of researchers. Swarm intelligence is the branch of artificial intelligence that deals with the implementation of intelligent systems by taking inspiration from the collective behavior of social insects and other societies of animals. Many meta-heuristic algorithms based on aggregative conduct of swarms through complex interactions with no supervision have been used to solve complex optimization problems. Data clustering organizes data into groups called clusters, such that each cluster has similar data. It also produces clusters that could be disjoint. Accuracy and efficiency are the important measures in data clustering. Several recent studies describe bio-inspired systems as information processing systems capable of some cognitive ability. However, existing popular bio-inspired algorithms for data clustering ignored good balance between exploration and exploitation for producing better clustering results. In this article, we propose a bio-inspired algorithm, namely social spider optimization (SSO), for clustering that maintains a good balance between exploration and exploitation using female and male spiders, respectively. We compare results of the proposed algorithm SSO with K means and other nature-inspired algorithms such as particle swarm optimization (PSO), ant colony optimization (ACO) and improved bee colony optimization (IBCO). We find it to be more robust as it produces better clustering results. Although SSO solves the problem of getting stuck in the local optimum, it needs to be modified for locating the best solution in the proximity of the generated global solution. Hence, we hybridize SSO with K means, which produces good results in local searches. We compare proposed hybrid algorithms SSO + K means (SSOKC), integrated SSOKC (ISSOKC), and interleaved SSOKC (ILSSOKC) with K means + PSO (KPSO), K means + genetic algorithm (KGA), K means + artificial bee colony (KABC) and interleaved K means + IBCO (IKIBCO) and find better clustering results. We use sum of intra-cluster distances (SICD), average cosine similarity, accuracy and inter-cluster distance to measure and validate the performance and efficiency of the proposed clustering techniques.

Keywords: Data clustering, Information processing systems, Swarm intelligence, Social spider optimization, K means clustering.

1 Introduction

Data clustering is one of the most frequently used mechanisms in data mining for summarizing large volumes of data sets [21]. The main objective of any data clustering approach is to minimize intra-cluster distances between data elements and maximize inter-cluster distances [9, 12, 22]. Data clustering can be done using two main clustering approaches, namely partitioned and hierarchical clustering [3]. The main advantage of partitioned clustering method is its capability of clustering large data sets [31]. It starts from an initial partitioning and relocates data objects by moving them from one cluster to another [20]. This method generally

***Corresponding author: Ravi Chandran Thalamala,** National Institute of Technology, Trichy, Tamil Nadu, India, e-mail: sirichandran007@gmail.com
A. VenkataSwamy Reddy and B. Janet: National Institute of Technology, Trichy, Tamil Nadu, India

requires that the number of clusters be preset by users. K means clustering is based on partitioned clustering approach. It minimizes the mean of squared distances from each data object to its nearest cluster centroid [24]. The reasons for popularity of K means include its linear time complexity, ease of interpretation, simplicity of implementation, speed of convergence and adaptability to work on sparse data [14].

Social spiders have an interesting and exotic collaborative behavior that provides advantages for survival [11]. They are capable of performing very complex tasks using a set of behavior rules and local information [19]. They show a tendency to live in colonies. In a colony, each member is capable of performing tasks such as predation, mating, web design and communication with other spiders [4]. Web is a main component of the colony. It acts as a common environment and a communication channel for all members [30]. It transmits important information such as trapped preys or mating possibilities to each member. Based on this local information, each member performs its cooperative behavior [16].

The performance of social spider optimization (SSO) algorithm for data clustering is compared with other data clustering methods. In summary, the present work makes the following contributions:

- a basic SSO algorithm for clustering data that avoids incorrect exploration and exploitation balance;
- three hybridized clustering algorithms that combine SSO and K means together to avoid the problem of getting stuck in local optima;
- to show the robustness of SSO, we have applied SSO on the standard data sets and got better results.

Section 2 describes related work on data clustering. In Section 3, the background of SSO is explained. We move on to SSO-based data clustering in Section 4. Experiments and results are explained in Section 5. We conclude with Section 6 in which scope of future work is specified.

2 Related Work

We will now outline some of the related work that has tackled different issues of data clustering using swarm intelligence (SI) in recent years. Forsati et al. [17] proposed an improved bee colony optimization (IBCO) algorithm with an application to data clustering. She introduced cloning and fairness concepts into BCO to make it more efficient for text document clustering. To overcome the problem of BCO algorithm in searching locally, she hybridized it with the K means algorithm to take advantage of the fine-tuning power of the widely used K means algorithm. The results showed that the proposed algorithm is 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 unknown or data objects are dynamically added or removed. Bharti and Singh [5] used chaotic map as a local search paradigm to improve exploitation capability of artificial bee colony (ABC) optimization. The experimental evaluation revealed very encouraging results in terms of the quality of solution and convergence speed. Cagnina et al. [6] proposed an efficient particle swarm optimization (PSO) approach to cluster data objects. 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. When the number of data objects is increased, a constant deterioration in the F measure values is observed with larger corpora. Karol and Mangat [25] proposed an evaluation of text document-clustering approach based on PSO. The proposed approach hybridizes fuzzy C means algorithm and K means algorithm with PSO. The performance of the proposed hybrid algorithm has been evaluated against traditional partitioning techniques. The authors concluded that the proposed algorithm deals better with overlapping nature of the data set. Shelokar et al. [34] proposed an algorithm that uses distributed agents to mimic how real ants find a shortest path from their nest to food source. Ahmadyfard and Modares [1] proposed an algorithm by combining PSO and K means algorithms to group a given set of data into a user-specified number of clusters. Elkamel et al. [15] proposed the communicating ants for clustering with backtracking strategy algorithm. It allows artificial ants to backtrack in their previous aggregation decisions. Jabeur [23] proposed a new firefly-based approach for wireless sensor network clustering. It has two phases: micro- and macro-clustering. In micro-clustering

phase, sensors self-organize into clusters. In macro-clustering phase, those clusters are polished by allowing aggregation of small neighboring clusters. Krishna and Murty [27] proposed a genetic K means algorithm (GKA) and found that it converges to the best known optimum corresponding to the given data and also observed that it searches faster than some of the other clustering algorithms. Krishnamoorthi and Natarajan [28] modified the traditional ABC algorithm with K means operator to optimize the clustering process and concluded that the proposed approach has upper hand over other methods. Coming back to SSO, it has not been applied to the clustering problem to the best of our knowledge.

2.1 Optimization Techniques

As there are some problems in partitioned clustering techniques, optimization techniques are proposed by researchers. They have been found to be successful in solving problems such as global optimization and multi objective optimization [2, 7, 13, 35]. In these techniques, an objective function that specifies the quality of clustering results is optimized by traversing through the solution space. We can use an optimization technique for clustering data or add optimization to the existing data clustering methods. An example for such optimization technique is SI. Different variants of SI have been proposed to either perform clustering independently or add to the existing clustering technique. Ant colony optimization (ACO) [34], particle swarm optimization (PSO) [26], and improved bee colony optimization (IBCO) [17] are the three main SI-based techniques that have been modeled and tested on different clustering problems thus far [3].

2.2 Evolutionary Techniques

When no technique provides an exact solution for an optimization problem or finding an exact solution is too computationally intensive, evolutionary techniques can be used to get a near-optimal solution. Evolutionary techniques are based on mechanisms inspired by biological evolution. The basic idea in evolutionary techniques is that with the help of evolutionary operators and a population of candidate solutions, convergence into a globally optimal solution can be attained [8]. An evolutionary technique mainly uses selection, element-wise average for intermediate recombination and mutation as the generic or evolutionary operators [36]. A fitness function is associated with each individual candidate solution to quantify the ability of the individual to survive and thrive in the search space [18]. Recombination takes two or more candidate solutions and produces two or more new candidate solutions. However, mutation takes one candidate solution and produces only one new candidate solution. Genetic algorithms is the most frequently used evolutionary technique in solving clustering problems [8]. Because of their random nature, evolutionary algorithms never produce an exact solution, but they will often produce a good solution if one exists.

3 Background of SSO

There are two fundamental elements of a social spider colony [11]. They are social members and communal web. The social members are divided into males and females. Spiders of female sex attract or dislike other spiders. Male spiders are classified into two classes, dominant and non-dominant (Figure 1). Dominant male spiders have better fitness than non-dominant male spiders. A dominant male mates with one or all females within a specific range to produce offspring.

Every spider has a weight based on the fitness value of the solution given by it. If fitness of spider is low, weight will be high in function minimization problem. Any spider whose weight is the largest of weights of all spiders is considered as globally best spider, s_{best}, and any spider whose weight is the smallest of weights of all spiders is considered as the worst spider, s_{worst} [33].

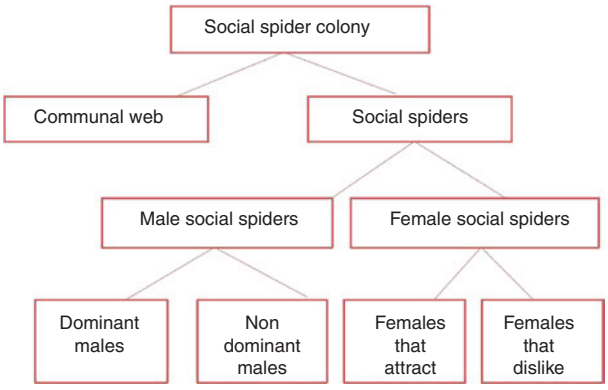

Figure 1: Elements of a Social Spider Colony.

Each spider is represented by a position in each dimension, weight and vibrations perceived from the other spiders. Spider position can be regarded as a candidate solution within the solution search space. The next position of a female spider depends on the nearest better spider and globally best spider, as shown in Figure 2. However, the next position of a dominant male spider depends on the nearest female spider, only as shown in Figure 3.

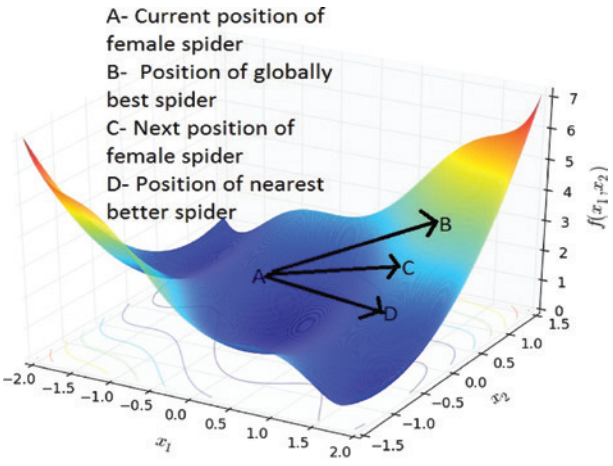

Figure 2: Generation of the Next Position of Female Spider (Reprinted from [33]).

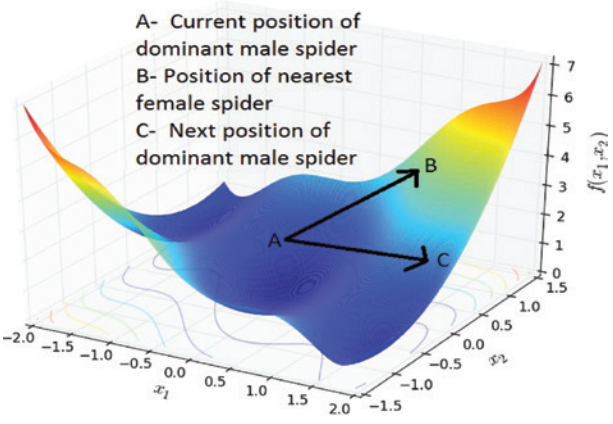

Figure 3: Generation of the Next Position of Dominant Male (Reprinted from [33]).

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 [11]. If the 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. The number of female spiders can be calculated using the following:

$$N_f = \text{floor}[(0.9 - \text{random}(0, 1) * 0.25) * N]. \tag{1}$$

Each spider position is randomly selected based on the upper and lower bounds of each dimension of objective function f as shown in the following:

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

where p_j^{high} and p_j^{low} are upper and lower bounds of the j^{th} dimension of objective function f to be optimized and $s_{i,j}$ is the initial position of spider s_i in the j^{th} dimension.

The weight w_i of each spider s_i represents quality of solution given by it. It can be calculated using the following:

$$w_i = (f(s_i) - \text{fit}_{\text{worst}}) / (\text{fit}_{\text{best}} - \text{fit}_{\text{worst}}), \tag{3}$$

where $f(s_i)$ is the fitness of spider s_i, fit_{best} is the minimum fitness in the population and $\text{fit}_{\text{worst}}$ is the maximum fitness (for minimization problem). The vibrations perceived as $vib_{i,j}$ by spider s_i from spider s_j can be calculated using the following:

$$vib_{i,j} = w_j * e^{-d^2}, \tag{4}$$

where d is the distance between spider s_i and spider s_j and w_j is the weight of spider s_j. Each spider s_i will perceive vibrations such as $vibc_i$ from the nearest better spider, $vibb_i$ from the globally best spider s_{best} and $vibf_i$ from the nearest female spider.

The female spiders attract or dislike other spiders irrespective of sex. 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 (threshold probability) TP, an attraction (+) movement is generated; otherwise, a repulsion (−) movement is produced. TP indicates the probability that a female spider attracts other spider. It is used to control attractions and repulsions of female spiders. It also controls the effect of vibrations perceived from globally best spider and nearest better spider on the next position of female spiders. If a female spider attracts or repulses only, a large portion of search area will be unexplored. TP is used to avoid this problem. If an attraction is generated, the next position of female spider f_i in the j^{th} dimension can be calculated using Equation (5). In this paper, we used the terms "spider" and "position of spider" interchangeably.

$$f_{i,j}^{\text{next}} = f_{i,j}^{\text{curr}} + \alpha * vibc_i * (s_{c,j} - f_{i,j}^{\text{curr}}) + \beta * vibb_i * (s_{\text{best},j} - f_{i,j}^{\text{curr}}) + \delta * (\gamma - 0.5) \tag{5}$$

If a repulsion movement is produced, the next position of the female spider f_i in the j^{th} dimension can be calculated using the following:

$$f_{i,j}^{\text{next}} = f_{i,j}^{\text{curr}} - \alpha * vibc_i * (s_{c,j} - f_{i,j}^{\text{curr}}) - \beta * vibb_i * (s_{\text{best},j} - f_{i,j}^{\text{curr}}) + \delta * (\gamma - 0.5). \tag{6}$$

In Equations (5) and (6), $f_{i,j}^{\text{curr}}$ is the current position of female spider f_i in the j^{th} dimension, $f_{i,j}^{\text{next}}$ is the next position of female spider f_i in the j^{th} dimension, $s_{\text{best},j}$ is the position of the globally best spider in the j^{th} dimension, $s_{c,j}$ is the position of the nearest better spider of female spider f_i in the j^{th} dimension, α, β, γ and δ are random numbers between 0 and 1, $vibc_i$ is the vibration perceived by spider f_i from its nearest better spider and $vibb_i$ are vibrations perceived by spider f_i from the globally best 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 the following:

$$r = \sum_{j=1}^{n} \frac{(p_j^{\text{high}} - p_j^{\text{low}})}{2 * n}, \tag{7}$$

where n is the number of dimensions present in objective function, p_j^{low} and p_j^{high} are the lower and upper bounds of the j^{th} dimension of the objective function, respectively.

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 m_i in the j^{th} dimension can be calculated using the following:

$$m_{i,j}^{\text{next}} = m_{i,j}^{\text{curr}} + \alpha * vibf_i * (f_{c,j} - m_{i,j}^{\text{curr}}) + \delta * (\gamma - 0.5), \tag{8}$$

where $m_{i,j}^{\text{next}}$ is the next position of dominant male spider m_i in the j^{th} dimension, $m_{i,j}^{\text{curr}}$ is the current position of dominant male spider m_i in the j^{th} dimension, $f_{c,i}$ is the position of nearest female spider f_c of dominant male spider m_i in the j^{th} dimension, α, γ and δ are random numbers between 0 and 1 and $vibf_i$ is the vibration perceived by spider m_i from its nearest female spider. The position of non-dominant male spider m_i in the j^{th} dimension can be calculated using the following:

$$m_{i,j}^{\text{next}} = m_{i,j}^{\text{curr}} + \alpha * (W - m_{i,j}^{\text{curr}}), \tag{9}$$

where $m_{i,j}^{\text{next}}$ is the next position of non-dominant male spider m_i in the j^{th} dimension, $m_{i,j}^{\text{curr}}$ is the current position of non-dominant male spider m_i in the j^{th} dimension and W is the weighted mean of male spiders.

Weighted mean W of male spiders in the j^{th} dimension can be calculated using Equation (10). If female spiders in the population are numbered from 1 to N_f, then male spiders will be numbered from N_f+1 to N_f+N_m, where N_m is the total number of male spiders in the population.

$$W = \frac{\sum_{h=1}^{N_m} m_h * w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}}. \tag{10}$$

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 wheel method. From Equations (5) to (9), it is clear that the next position of female spiders is influenced only by positions of the globally best and nearest better spiders. The next position of dominant male spiders is dependent only on the position of the nearest female spider. Because of this, SSO can search solution space in different directions at the same time. Let S_d be a dominant male spider, F be the set of all female spiders within the range of mating operation and T be the set of all spiders which are participating in mating operation. T can be calculated using the following:

$$T = S_d \ U \ F. \tag{11}$$

s_{new}, the position of the resultant spider of mating operation, can be calculated using the roulette wheel method, as shown in Equation (12). Let t be total number of spiders in T. Let T_i be the position of the i^{th} spider in T and W_i be its weight.

$$s_{\text{new}} = \frac{\sum_{i=1}^{t} T_i * w_i}{\sum_{i=1}^{t} w_i}. \tag{12}$$

Before mating operation, each dominant male spider identifies all female spiders whose fitness is less than or equal to r, which is the range of mating operation. The dominance of male spiders and vibrations of female spiders play an important role in SSO optimization.

The most popular swarm algorithms like PSO, ABC and ACO have critical flaws such as incorrect exploration and exploitation balance and premature convergence [10]. SSO divides the entire population into two agent categories, namely female and male spiders. Efficient exploration is achieved through the female

spiders, and extensive exploitation is achieved through the male spiders. As SSO has the capacity of finding a good balance between exploration and exploitation, it can be used to find the global optimal solution.

4 SSO-Based Data Clustering

The SSO algorithm is a population-based, nature-inspired, meta-heuristic evolutionary optimization technique. It is quite similar to how social spiders in nature are cooperative to one and another. In SSO algorithm, a spider simulates a candidate solution for the given optimization problem. The fitness of spider represents the goodness of solution. The web simulates the entire solution space. The behavior rules of spiders are simulated to find the next positions of the spiders. The mating operation is simulated to get the position of the new spider.

In SSO-based data clustering, each spider represents a collection of clusters of data objects. The algorithm starts with initializing each spider with K randomly chosen data objects, where K is number of clusters to be formed. These K data objects in each spider s_r will be treated as K initial centroids. Each data object in the data set is associated with exactly one of these K centroids based on distance measure. Then, we calculate fitness and weight of each spider using Equations (13) and (3), respectively. The fitness of each spider s_r is the average distance between data objects and cluster centroids. Assume that clusters to be formed are C_1, C_2, C_3, …, C_K. Then, the fitness fit_r of spider s_r can be calculated using the following:

$$\mathrm{fit}_r = \frac{\sum_{i=1}^{K} \frac{\sum_{j=1}^{n_i} \mathrm{distance}(\mathrm{centroid}_i, \mathrm{doc}_j)}{n_i}}{K} = f(s_r = \{C_1, C_2, C_3, \ldots, C_K\}), \tag{13}$$

where $\mathrm{centroid}_i$ is the centroid of cluster C_i, doc_j is the j^{th} data object present in cluster C_i, n_i is the number of data objects in cluster C_i, K is the number of clusters in each spider and distance is the distance measure function that takes two data object vectors [32].

Algorithm 1: SSO data clustering algorithm.

```
1:  procedure SSO clustering (Inputs: data set of data objects; D, number of clusters to be formed; K, maximum number of
    iterations; Max, threshold probability; TP, number of spiders; N, Output: clusters of relevant data objects)
2:      Compute N_f as the number of female spiders and N_m as the number of male spiders
3:      Assign K randomly chosen data objects for each spider in the population
4:      Initialize iteration with 1
5:      while iteration ≤ Max do
6:          Find the Euclidian distance between each data object and each centroid and associate the data object to the
            nearest cluster centroid
7:          Find the average distance between data objects and their cluster centroids in each spider and take it as fitness of
            spider, as specified by Equation (13)
8:          Find the best and worst spiders and then find the weight of each spider using Equation (3)
9:          Move female spiders to their next positions using Equations (5) and (6)
10:         Move male spiders to their next positions using Equations (8) and (9)
11:         Perform mating operation of each dominant male spider within the specified range of mating and then replace the
            worst spider with a new spider if the weight of the new spider is greater than the weight of the worst spider
12:         Increment iteration by 1
13:     end while
14:     Return spider with best fitness
15: end procedure
```

Algorithm 2: SSOKC clustering algorithm.

```
1: procedure SSOKC clustering (Inputs: data set of data objects; D, number of clusters to be formed; K, maximum number of
   iterations; Max, threshold probability; TP, number of spiders; N; Output: clusters of relevant data objects)
2:    Execute SSO clustering for 50 to 100 iterations
3:    Inherit clustering results from SSO as K initial cluster centroids for K means clustering process
4:    Start K means clustering process until convergence is achieved
5: end procedure
```

The smaller the average distance between data objects and the cluster centroid, the more compact the clustering solution is [32]. Hence, we consider data clustering problem as a 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 the stoping criteria are met. SSO-based data clustering is summarized in Algorithm 1. It returns the spider with minimum average distance between the data objects and their centroids.

4.1 Hybridized SSO-Based Data Clustering

In SSO, the vibrations perceived from the globally best spider contribute to exploration. The vibrations perceived from the nearest better and nearest female spiders contribute to exploitation. However, if the distance between the current spider and its nearest better spider (or nearest female spider) is high, the vibrations perceived from the nearest better spider (or nearest female spider) contribute to exploration. Thus, there is some scope of imbalance between exploration and exploitation. SSO is powerful in exploring search space. K means algorithm is powerful in exploitation of local neighborhood. To get the right balance between global wide exploration and local neighborhood exploitation during the search process, we proposed SSOKC. It contains the functionalities of both SSO and K means. Solutions generated by SSO are improved locally using K means. SSOKC algorithm includes two modules, namely SSO module and K means module. At the initial stage, the SSO module is used for discovering the vicinity of optimal solution by a global search. The global search of SSO produces centroids of K clusters. These centroids are then passed to K means module for refining and generating the final optimal clustering solution. The process is summarized in Algorithm 2. We took compositions of SSO and K means in three different combinations which combine the global searching power of SSO and local refining capability of K means to maintain a right balance between exploitation and exploration.

- SSOKC: Initially, SSO algorithm is executed for 50–100 iterations, and the result is given as an input to K means algorithm that refines the result.
- Integrated SSOKC (ISSOKC): After every iteration of SSO, K means is executed using thecurrent best solution of SSO as initial seed. If fitness of the solution given by K means is better than that of the current best solution of SSO, the solution of K means replaces the current best solution of SSO.
- Interleaved SSOKC (ILSSOKC): After every n iterations of SSO, K means is executed by using current best solution of SSO as initial seed. If fitness of the solution given by K means is better than that of current best solution of SSO, solution of K means replaces the current best solution of SSO.

5 Experiments and Results

5.1 Data Sets

The proposed clustering approaches are applied on the data sets namely Iris, Glass, Ruspini, Vowel, Wine and Wisconsin Breast Cancer, collected from UCI Irvine Machine Learning Repository [29]. The attributes of

all data objects are of numeric datatype. Average cosine similarity, average inter-cluster distance and accuracy are used as metrics in each of the algorithms. No parameter setting is required for K means.

5.2 Evaluation of SICD

We use SICD also to measure and validate the performance and efficiency of clustering techniques. Lower SICD value indicates higher clustering quality.

Assume that data objects are dataoject$_1$, dataobject$_2$, dataoject$_3$, …, dataoject$_n$. The clusters to be formed are C_1, C_2, C_3, …, C_K and their centroids are centroid$_1$, centroid$_2$, centroid$_3$, …, centroid$_K$. Then, SICD is calculated as

$$\text{SICD} = \sum_{i=1}^{K} \sum_{j=1}^{n} \text{distance}(\text{centroid}_i, \text{dataobject}_j), \tag{14}$$

where distance (centroid$_i$, dataobject$_j$) is the Euclidian distance between the centroid$_i$ and the data object dataobject$_j$ if dataobject$_j$ is placed in cluster C_i; otherwise, zero.

5.3 Experimental Setup

The proposed clustering approaches are applied on the six data sets summarized in Table 1. The Euclidian distance function is used in each algorithm to find the distance (similarity) between any two data objects. We have noticed that K means clustering algorithms can converge to a stable solution within 20–30 iterations when applied to most data sets. We used Intel® Xeon® CPU E3 1270 v3 with 3.50-GHz processor, RAM of 160-GB capacity, Windows 7 Professional Operating System and Java Run Time Environment of version 1.7.0.51 in our research.

6 Results and Discussion

The modeled behaviors of female and male spiders explicitly avoid their concentration at current best positions. This fact avoids the critical flaws such as premature convergence and incorrect exploration and exploitation balance. In all the tables, we specified best results in bold font. We found that as we increase the number of iterations, accuracy, average cosine similarity, and F measure are also increased in SSO-based data clustering. As we increase the number of iterations, more and more spiders will be replaced by better newly generated spiders of mating operation, yielding higher cosine similarity. The results of SSO clustering are specified in Table 2. The accuracy of a clustering method is the ratio of the sum of true positives and true negatives to the total number of data objects.

Table 3 shows how the SICD changes during iterations 50, 100, 150, 200, 250 and 300 in SSO algorithm. Initially, as we consider all spiders irrespective of their fitness values, the result of SICD is high. However, as we increase the number of iterations, more and more worst spiders are replaced by newly generated better spiders from the mating operation. Therefore, as we increase the number of iterations, population will have only better spiders, resulting in low SICD. Figure 4 shows the effect of the parameter TP on accuracy.

Table 1: Summary of Data Sets.

	Iris	Glass	Vowel	Wine	Cancer	Ruspini
Number of data objects	150	214	871	178	683	75
Number of classes	3	6	6	3	2	4
Number of attributes	4	10	3	13	9	2

Table 2: SSO Clustering.

Data set	SICD	Average cosine similarity	F measure	Accuracy
Iris	97.43	0.9468	0.9433	94.66
Glass	213.23	0.9948	0.8238	86.5264
Vowel	149,403.88	0.8123	0.8193	88.7433
Wine	16,287.63	0.9990	0.7295	81.6479

Table 3: SICD Variation with Number of Iterations: SSO Clustering.

Data Set	50 Iterations	100 Iterations	150 Iterations	200 Iterations	250 Iterations	300 Iterations
Iris	99.87	99.68	98.75	97.66	97.59	**97.43**
Glass	221.23	221.07	218.25	216.68	214.03	**213.23**
Vowel	150,925	150,726	150,250	150,003	149,725	**149,403.88**
Wine	16,305	16,300	16,297	16,295	16,289	**16,287.63**

In all the tables, best results have been specified in bold font.

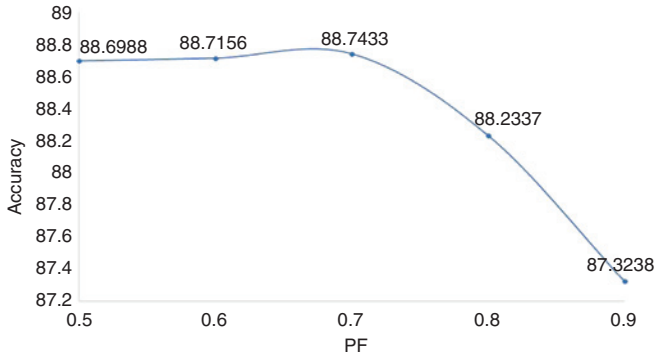

Figure 4: Effect of TP on Accuracy: SSO (Vowel Data Set).

We found that when TP is less than or equal to 0.7, better results are produced, but when TP reaches closer to 1, the results are not better due to reduced solution space. When probability that female spider repulses another spider reaches closer to zero, female spider behavior will be defined by only attraction, resulting in reduced solution space and comparatively poor clustering results.

We also checked the convergence of SSO in the wine data set. In Figure 5, SICD stays the same after 300 iterations. This implies that convergence is achieved.

We found that when Euclidian distance function is used in SSO clustering method, better average cosine similarity and accuracy are produced when compared with Manhattan distance function, as shown in Table 4. The reason is that Euclidian distance function is not influenced by very small differences in corresponding attribute values unlike Manhattan distance function. In other words, the data objects that have very small Euclidian distance will more likely be placed in same cluster.

Table 5 shows how clusters are formed when we use SSO clustering. We also measured inter-cluster distances when SSO clustering method is used. Inter-cluster distance can be defined as sum of the square distance between each cluster centroid. Clustering technique should maximize this inter-cluster distance. The number of data objects in each cluster is also specified.

To show the adaptability of SSO clustering for the change in configuration of data sets, we compare results of random centroids, random data sets and 10×10 cross-validation techniques. Table 6 uses the Ruspini data set and shows the results of random centroids, random data sets and 10×10 cross-validation techniques using different measures such as inter-cluster distance and SICD with mean and standard deviation. In

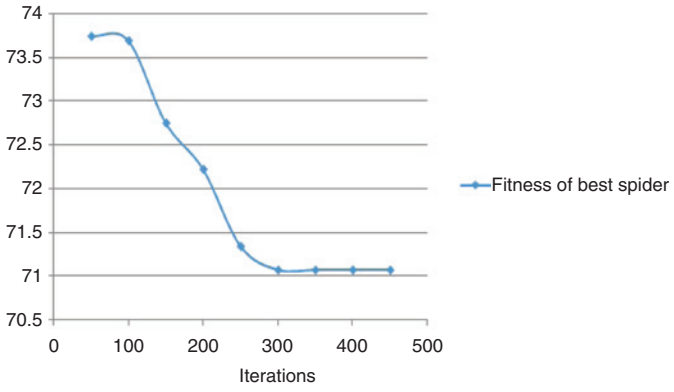

Figure 5: Convergence Analysis for Wine Data Set: SSO.

Table 4: Effect of Distance Functions on SSO.

Data Set	Euclidian distance function		Manhattan distance function	
	Accuracy	Average cosine similarity	Accuracy	Average cosine similarity
Iris	**94.6666**	**0.9468**	94.0000	0.9398
Glass	**86.5264**	0.9948	84.4444	**0.9949**
Vowel	**88.7433**	**0.8123**	85.9259	0.7822
Wine	**81.6479**	**0.9990**	80.8888	0.9990
Ruspini	**100.0000**	**0.9907**	99.1666	0.9907

In all the tables, best results have been specified in bold font.

Table 5: SSO Cluster Distribution.

Data set	Data per cluster	Average intra-cluster distance	SICD	Inter-cluster distance
Iris	50, 46, 54	0.6495	97.43	5.597
Glass	9, 42, 66, 16, 31, 50	0.9964	213.23	272.5176
Wine	61, 49, 68	91.597	16,287.63	325.9995
Ruspini	15, 20, 23, 17	11.205	840.3750	1285.3700

Table 6: Cross-Validation of Ruspini Data Set: SSO.

	Random centroids	Random data sets	Cross-validation
Sum of intra-cluster distance			
Mean	851.8166	855.42	**850.24**
Standard deviation	**9.6104**	11.09	11.27
Best	840.37	840.22	**836.18**
Worst	865.23	872.11	**868.49**
Inter-cluster distance			
Mean	1264.67	1263.95	**1281.17**
Standard deviation	**13.94**	14.28	14.27
Best	1285.37	1283.94	**1304.71**
Worst	1247.17	1244.34	**1265.73**

In all the tables, best results have been specified in bold font.

random centroids technique, the centroids were randomly selected. In random data sets technique, the data was shuffled to make it random. The first column specifies the results of the random centroids technique. The second column reports the results of random data sets technique, and the last column shows 10×10

Table 7: SICD Comparison: Clustering Algorithms

Data set	K means		PSO		IBCO		ACO		SSO	
	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
Iris	106	97	103	**96**	97	97	97	97	**97**	97
Glass	260	215	291	271	225	214	NA	NA	**224**	**213**
Vowel	159,242	149,422	168,477	163,882	150,881	149,466	NA	NA	**150,794**	**149,403**
Wine	18,161	16,555	16,311	16,294	16,460	16,460	16,530	16,530	**16,304**	**16,287**

In all the tables, best results have been specified in bold font.

cross-validation when the centroids were randomly initialized. It is found that the results of the three techniques are more or less the same. This indicates the stability of the SSO algorithm for the change in configuration of data sets. It is also found that the 10×10 cross-validation technique produced slightly better results than the other techniques with respect to inter-cluster distance and SICD. The reason is that it takes relatively small number of data instances (i.e. 10 data instances) of each class as input unlike the other techniques.

6.1.1 Comparison to Other Clustering Methods

We conducted experiments to compare the performance of the proposed algorithms with K means, PSO-based clustering [26], ACO-based clustering [34], and IBCO clustering [17]. As shown in Table 7, the SSO sclustering method produced minimal SICD value for all data sets.

- PSO-based clustering: It starts with a set of candidate solutions for the clustering problem. The solutions are considered as particles. Each particle has position and velocity. The movement of a particle in solution space is influenced by its locally and globally best positions thus far of the swarm. The particles move toward best solution.
- ACO-based clustering: The ACO clustering simulates the way real ants find the shortest path between a food source and their nest. The communication among ants happens by means of pheromone trails. They exchange information about the path to be followed. If the path contains more ants traces, it becomes more attractive. The collective behavior of ants enable them to find the shortest path to the food source.
- IBCO clustering: A major shortcoming of BCO clustering is the imbalance between exploration and exploitation. The exploratory power of BCO has been increased with fairness and cloning concepts in IBCO clustering.

We compare SSO, K means, PSO and SSOKC using accuracy and found that SSOKC outperforms the other three clustering methods due to its capability of exploring a wide search space to produce optimal solution. We calculated the accuracy and its standard deviation for the clustering methods and found that SSOKC produced the best clustering accuracy, as shown in Table 8.

Table 8: Comparison of Accuracy of Clustering Methods.

Data set	PSO		SSO		K means		SSOKC	
	Accuracy	SD	Accuracy	SD	Accuracy	SD	Accuracy	SD
Iris	93.9245	1.76	94.6666	1.56	94.0000	1.53	**94.6666**	**1.25**
Glass	84.2679	3.91	86.5264	10.63	82.7881	5.79	**88.1651**	**0.32**
Vowel	88.1947	0.73	88.7433	**0.10**	88.8238	2.00	**89.4268**	0.12
Wine	81.2734	1.59	81.6479	0.00	80.1498	0.79	**82.1498**	**0.00**
Ruspini	100.0000	0.00	100.0000	0.00	88.0000	2.50	**100.0000**	**0.00**

SD, Standard deviation. In all the tables, best results have been specified in bold font.

6.1.2 Comparison to Other Hybrid Clustering Methods

We compare the proposed hybrid algorithms with hybrid models such as KPSO, KGA, KABC and IKIBCO. The clustering results of these existing hybrid models are taken from [17]. To evaluate the quality of clustering obtained by these hybrid algorithms, we used SICD as a metric. We found that ILSSOKC outperformed all other hybrid models as shown in Table 9. The algorithmic parameters used for each clustering algorithm is reported in Table 10.

- KPSO: There are two phases in this algorithm. In the first phase, K-means algorithm is used to find a solution for the clustering problem. The resultant solution will be treated as one particle in PSO. The remaining particles are initialized randomly. Then PSO clustering is applied.
- KGA: In GA, the child chromosomes are obtained from parents chromosomes using the costly fitness function or the expensive cross-over or both. In KGA, the cross-over function will be replaced by K means operator.
- KABC: In the ABC algorithm, the honey bees are classified as employed, onlooker, and scout bees. The employed bees search for the food source and pass that information to onlooker bees. The onlooker bees will select the food souce that has higher quality. The employed bee whose food source has been eliminated becomes a scout and starts to search for finding a new food source. KABC optimises the clustering process using ABC algorithm with K means operator.
- IKIBCO: The results of K means are passed to IBCO clustering and then IBCO continues its execution.

6.1.3 Comparison of Proposed Algorithms with Respect to CPU Usage Time

We compare the proposed algorithms on basis of CPU usage time (best value) during the clustering process. Table 11 depicts CPU usage (in seconds) of the proposed algorithms. It is evident that ISSOKC took more time

Table 9: SICD Comparison: Hybrid Clustering Algorithms.

Data set	KPSO		KGA		KABC		IKIBCO		ISSOKC		ILSSOKC	
	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
Iris	96.76	96.66	97.1	96.10	96.29	96.19	**95.14**	**95.10**	95.49	95.32	95.45	95.22
Glass	221.55	213.37	221.7	215.7	221.89	215.3	221.35	214.71	220.84	213.02	**217.35**	**212.86**
Vowel	150,990	149,486	150,992	149,556	150,903	149,498	150,892	149,473	150,744	149,400	**150,169**	**149,389**
Wine	16,296	16,292	16,298	16,295	16,296	16,292	16,294	16,292	**16,291**	16,283	16,294	**16,283**

In all the tables, best results have been specified in bold font.

Table 10: Values of Parameters for Different Clustering Algorithms.

SSO		PSO		IBCO		ACO	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
No. of spiders	50	Population	100	No. of bees	20	No. of ants	50
TP	0.7	Min and max inertia	0.7	No. of iterations	[1, 1000]	Probability for max trial	0.98
No. of iterations	[50, 300]	Acceleration factor (c1)	2	γ	[0, 1]	Local search probability	0.01
$\alpha, \beta, \gamma, \delta$	[0, 1]	Acceleration factor (c2)	2	NA	NA	Evaporation rate	0.01
NA	NA	No. of iterations	[1, 1000]	NA	NA	No. of iterations	[1, 1000]
NA	NA	V_{min}	−0.05	NA	NA	NA	NA
NA	NA	V_{max}	0.05	NA	NA	NA	NA

TP, Threshold probability.

Table 11: CPU Time-Elasped Comparison: Proposed Algorithms (Vowel Data Set).

Algorithm	50 Iterations	100 Iterations	150 Iterations	200 Iterations	250 Iterations	300 Iterations
SSO	**22.09**	**39.47**	**45.55**	**52.00**	**60.26**	**64.77**
ISSOKC	43.96	52.31	67.88	74.88	79.63	86.80
ILSSOKC	32.09	40.17	55.99	63.17	68.00	75.00

In all the tables, best results have been specified in bold font.

to complete the execution process when compared with the other two algorithms. The reason is that after each iteration of SSO, K means has to be executed.

7 Conclusion and Future Work

Thus far, SSO has not been applied to the data clustering problem. We experimented some methods of applying SSO to solve the clustering problem. We presented our work where SSO was independently applied to the clustering problem. We then described how it can be hybridized with K means clustering to improve accuracy, cosine similarity, SICD and inter-cluster distance. The comparison of results showed that ILSSOKC is the best clustering method when compared with KPSO, KGA, KABC, IKIBCO and ISSOKC clustering methods. We showed how parameters like TP and random variables affect the clustering results. We also showed the effect of distance measure functions like Euclidian and Manhattan distance functions on SSO clustering. Our work leaves a few unexplored directions. We used only static structure in the implementation. However, when the number of clusters is unknown or the data objects are added or removed dynamically, a dynamic structure is needed. The dynamic problem is much more challenging and requires careful investigations. Future work includes generalization of the clustering method so that it can be applied on multimedia data. It also includes analysis of the applicability of the clustering method on big data.

Bibliography

[1] A. Ahmadyfard and H. Modares, Combining PSO and k-means to enhance data clustering, in: *Telecommunications, 2008. IST 2008. International Symposium on*, pp. 688–691, IEEE, Tehran, Iran, 2008.

[2] S. Alam, G. Dobbie and P. Riddle, An evolutionary particle swarm optimization algorithm for data clustering, in: *Swarm Intelligence Symposium, 2008*, pp. 1–6, IEEE, 2008.

[3] S. Alam, G. Dobbie and S. Ur Rehman, Analysis of particle swarm optimization based hierarchical data clustering approaches, *Swarm Evol. Comput.* **25** (2015), 36–51.

[4] L. Aviles, Sex-ratio bias and possible group selection in the social spider *Anelosimus eximius*, *Am. Nat.* **128** (1986), 1–12.

[5] K. K. Bharti and P. K. Singh, Chaotic gradient artificial bee colony for text clustering, Fourth International Conference of Emerging Applications of Information Technology, pp. 337–343, IEEE, Kolkata, India, 2014.

[6] L. Cagnina, M. Errecalde, D. Ingaramo and P. Rosso, An efficient particle swarm optimization approach to cluster short texts, *Inform. Sci. (Ny)* **265** (2014), 36–49.

[7] C.-Y. Chen and F. Ye, Particle swarm optimization algorithm and its application to clustering analysis, in: *Networking, Sensing and Control, 2004 IEEE International Conference on*, 2, pp. 789–794, IEEE, Tehran, Iran, 2004.

[8] K. J. Cios, W. Pedrycz and R. W. Swiniarski, *Data mining and knowledge discovery*, Springer Science & Business Media, 1998.

[9] P. Cudré-Mauroux, S. Agarwal and K. Aberer, Gridvine: an infrastructure for peer information management, *IEEE Internet Comput.* **11** (2007), 36–44.

[10] E. Cuevas and M. Cienfuegos, A new algorithm inspired in the behavior of the social-spider for constrained optimization, *Expert Syst. Appl.* **41** (2014), 412–425.

[11] E. Cuevas, M. Cienfuegos, D. Zaldvar and M. Pérez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, *Expert Syst. Appl.* **40** (2013), 6374–6384.

[12] L. F. da Cruz Nassif and E. R. Hruschka, Document clustering for forensic analysis: an approach for improving computer inspection, *IEEE Trans. Inf. Forensics Security* **8** (2013), 46–54.
[13] S. Das, A. Chowdhury and A. Abraham, A bacterial evolutionary algorithm for automatic data clustering, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pp. 2403–2410, IEEE, Trondheim, Norway, 2009.
[14] I. S. Dhillon and D. S. Modha, Concept decompositions for large sparse text data using clustering, *Machine Learning* **42** (2001), 143–175.
[15] A. Elkamel, M. Gzara and H. Ben Abdallah, A bio-inspired hierarchical clustering algorithm with backtracking strategy, *Appl. Intel.* **42** (2015), 174–194.
[16] C. Eric and K. S. Yip, Cooperative capture of large prey solves scaling challenge faced by spider societies, in: *Proceedings of the National Academy of Sciences of the United States of America*, 105, pp. 11818–11822, Washington, USA, 2008.
[17] R. Forsati, A. Keikha and M. Shamsfard, An improved bee colony optimization algorithm with an application to document clustering, *Neurocomputing* **159** (2015), 9–26.
[18] D. E. Goldberg, *Genetic algorithms in search optimization and machine learning*, 412, Addison-Wesley Reading, Menlo Park, CA, 1989.
[19] D. Gordon, The organization of work in social insect colonies, *Complexity* **8** (2003), 43–46.
[20] M. Gupta and R. Jain, A performance evaluation of SMCA using similarity association & proximity coefficient relation for hierarchical clustering, *Int. J. Eng. Trend. Technol. (IJETT)* **15** (2014), 354.
[21] M. T. Hassan, A. Karim, J.-B. Kim and M. Jeon, Document clustering by discrimination information maximization, *Inf. Sci.* **316** (2015), 87–106.
[22] Y. Ioannidis, D. Maier, S. Abiteboul, P. Buneman, S. Davidson, E. Fox, A. Halevy, C. Knoblock, F. Rabitti, H. Schek, G. Weikum, Digital library information-technology infrastructures, *Int. J. Digit. Lib.* **5** (2005), 266–274.
[23] N. Jabeur, A firefly-inspired micro and macro clustering approach for wireless sensor networks, *Procedia Comput. Sci* **98** (2016), 132–139.
[24] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman and A. Y. Wu, The analysis of a simple k-means clustering algorithm, in: *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, pp. 100–109, ACM, Clear Water Bay, Hong Kong, 2000.
[25] S. Karol and V. Mangat, Evaluation of text document clustering approach based on particle swarm optimization, *Open Comput. Sci.* **3** (2013), 69–90.
[26] R. C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in: *Proceedings of the sixth international symposium on micro machine and human science,* Vol. 1, pp. 39–43, Nagoya, Japan, 1995.
[27] K. Krishna and M. N. Murty, Genetic K-means algorithm, *IEEE Trans. Syst. Man. Cybern. B (Cybern.)* **29** (1999), 433–439.
[28] M. Krishnamoorthi and A. M. Natarajan, ABK-means: an algorithm for data clustering using ABC and K-means algorithm, *Int. J. Comput. Sci. Eng.* **8** (2013), 383–391.
[29] M. Lickman, *UC irvine machine learning repository*, 2013.
[30] S. Maxence, Social organization of the colonial spider *Leucauge* sp. in the Neotropics: vertical stratification within colonies, *J. Arachnol.* **39** (2010), 446–451.
[31] S. K. Popat and M. Emmanuel, Review and comparative study of clustering techniques, *Int. J. Comp. Sci. Inform. Technol.* **5** (2014), 805–812.
[32] T. Ravi Chandran, A. V. Reddy and B. Janet, A social spider optimization approach for clustering text documents, in: *Proceedings of the 2nd International Conference on Advances in Electrical and Electronics, Information Communication and Bio Informatics*, pp. 22–26, IEEE, 2016.
[33] T. Ravi Chandran, A. V. Reddy and B. Janet, Text clustering quality improvement using a hybrid social spider optimization, *Int. J. Appl. Eng. Res.* **12** (2017), 995–1008.
[34] P. S. Shelokar, V. K. Jayaraman and B. D. Kulkarni, An ant colony approach for clustering, *Anal. Chim. Acta* **509** (2004), 187–195.
[35] D. W. Van der Merwe and A. P. Engelbrecht, Data clustering using particle swarm optimization, in: *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, 1, pp. 215–220, IEEE, Canberra, ACT, Australia, 2003.
[36] X. S. Yang and Z. W. Geem, *Music-inspired harmony search algorithm: theory and applications*, Springer, Part of the Studies in Computational Intelligence book series (SCI, volume 191), 2009.

Bionotes

Ravi Chandran Thalamala
National Institute of Technology, Trichy,
Tamil Nadu, India,
sirichandran007@gmail.com

Ravi Chandran Thalamala received his postgraduate degree from Nagarjuna University, Guntur, India, in 2000. He is currently a PhD candidate at the Department of Computer Applications, National Institute of Technology, Trichy, India. His research interests are in the areas of bio-inspired algorithms, data mining, artificial intelligence and software engineering.

A. Venkata Swamy Reddy
National Institute of Technology, Trichy,
Tamil Nadu, India

A. Venkata Swamy Reddy received his PhD degree from the Indian Institute of Sciences, Bangalore, India, in 1985. He is a professor at the Department of Computer Applications, National Institute of Technology, Trichy, India. He has more than 30 years of research experience. His research interests are in the areas of design and analysis of algorithms, computer networks, data mining, operating systems and theoretical computer science. He has published more than 30 articles in international journals and more than 25 papers in proceedings of international conferences.

B. Janet
National Institute of Technology, Trichy,
Tamil Nadu, India

B. Janet received her undergraduate degree in BSc major in physics with distinction from Holy Cross College, Trichy, in 1999, from the Bharathidasan University, Trichy, India, and postgraduate degree in master of computer applications in 2002 from Bishop Heber College, with a university third rank from Bharathidasan University, Trichy. She started her research in information retrieval with a master of philosophy in computer science from Alagappa University, Karaikudi, India, in 2005. She was awarded her PhD degree in 2012 by the National Institute of Technology, Trichy. Since 2002, she has been a professional facilitator of students. Presently, she is an assistant professor at the Department of Computer Applications, National Institute of Technology, Trichy, Tamil Nadu, India. She has 15 years of teaching experience, which includes experiments on activity-based learning, learner-centric teaching and flipped classrooms. She has 9 years of research experience, with more than 15 research papers to her credit. Her research interests include information retrieval and information security.