CrossMark

An improved neighbor cooperation using adjacent node cooperation in peer to peer networks

S. Raju[1] · M. Chandrasekaran[2]

Abstract

The peer-to-peer (P2P) computation is that resource of the computer where sharing takes place through direct exchange. A Computational optimization is now getting much more prevalent in various fields in which there are simple solutions to a problem that are analytical. The swarm intelligence is defined as the behaviour of the artificial, natural, self-organized and decentralized systems. The multi-objective optimization (MOO) also known as the vector optimization is optimized in place of a single objective. The artificial fish swarm optimization (AFSA) has a search capacity that is global and also has a strong robustness being insensitive to the initial values. In this study the MOO has been hybridized by a simulated annealing using a k-means clustering and the AFSA by means of using the cooperation of neighbours. This multi-objective system helps in easing the difficulties of being sensitive to the initial solutions. This paper introduces an AFSA multi-objective system that promises to improve the neighbour cooperation in the application of P2P network.

1 Introduction

Peer-to-peer (P2P) networks [1] may also link various end hosts (known as peers) in a very ad-hoc way. The P2P networks are particularly made use in applications of file sharing that enables the peers to share all the digitalized files like audio, video, electronic books and so on. In recent times much more advanced applications like real-time conferences, media streaming, and online gaming have also been included in this. Unlike many of the traditional networks that have server client topology and the files are available in the data center every peer is a client and also a server in the P2P networks. This model is very convenient owing to the classical ability to search for the files throughout the network by means of using semi-centralized servers that are shown as a centralized node that leaves the network vulnerable.

In case of a P2P network, the nodes to communication will be able to communication with neighbors with no central coordination. By means of exchange of data this network performs a cooperative positioning: where each node gets helped by the neighbors for computing this position.

Optimization is that process that has been defined as: Finding of the value in the variables which will either minimize or sometimes maximize the functions of an objective at the same time satisfying such constraints. The problems in optimization have been centered based on three factors: (1) the objective function that is either minimized or maximized. (2) One set of variables or unknowns which affect this objective function (3) constraints allowing unknowns taking on some values at the same time excluding others [2].

In most problems of optimization there are either one or more local solutions and it does become more important in choosing a proper method of optimization which will not be greedy and will look in its neighborhood for an ideal solution. This can mislead the process of search and will leave this stuck in a local solution. But the problem of optimization will have a mechanism for balancing between that of the local as well as the global search. There are several methods that have been used for solving the problems of optimization in mathematical as well as combinatorial type. In case the

✉ S. Raju
sraju.mahe@yahoo.com; rasakudil@gmail.com

M. Chandrasekaran
drmcs123@yahoo.com

1   Mahendra Engineering College, Namakkal, Tamilnadu, India

2   Government Engineering College, Bargur, Tamilnadu, India

problem gets difficult or if the space becomes large this can be quite difficult to solve. Owing to this there are many methods of meta-heuristic optimization developed to solve them.

Multi-objective optimization can meet the system needs that trade-off between conflicting objectives than that of single optimization problems. This normally includes many goals that are conflicting and its outcome is achieved by compromising the goal. Presently there are many approaches that are of multi-objective including traditional mathematical approaches like Goal programming and intelligent algorithms like Ant Colony algorithm, Particle Swarm Optimization, and Genetic Algorithms.

The k-means protocol completely depends on the Euclidian distances and residual energies that are the basis of the selection of cluster heads. Therefore, all the nodes collect information on the node id, the position, and the nodes' residual energy and store the information in the central node. Once this information is procured the clustering algorithm (k-mean) is performed [3]. This technique will distribute the dataset into different clusters having similar features [4]. In order to solve this problem of clustering, the cluster number that fits into a dataset is identified and the objects for these are duly assigned.

The K means clustering is that well studied and popular technique of data analysis. A standard version will assume that data that is available in a particular location is easy to get. But if the sources of data get distributed in a large P2P network [5], the collection of such data in a central location even before that of clustering will not be attractive or practical. There are several exciting applications in K-means clustering for the data that is distributed over that of a P2P network and these are now highly scalable, efficient, and is desired.

The Heuristics makes use of trial and error for producing solutions that are acceptable which is a common problem within a practical and reasonable time [6]. The main complexity of this problem will make it very difficult in searching for a solution. The aim here is to identify a solution that is acceptable inside a timescale. The Meta-heuristic will be formally defined as a process of iterative generation for guiding a subordinate heuristic by means of intelligently combining the concepts in exploration as well as the exploitation in its search space and the strategies of learning will be used for structuring the information to identify solutions that are efficient and near-optimal.

In this work, the simulated annealing (SA) can be the answer to this. This is not to fool with false minima and is very easy in its implementation. Also as it does not need a mathematical model, it may also be used in solving a wide range of problems. The Simulated annealing is that which imitates the process of annealing that is used in metallurgy. When a substance goes through the annealing process it is heated for some time until it attains a point of fusion for

liquefying it and is later cooled down to solid again. The lasting properties of this depend on the schedule of cooling that is applied; if cooling happens quickly it is easily broken down into an imperfect structure and if it is slow the structure is strong.

Simulated Annealing [7] can be both a generic and probabilistic meta-algorithm for the problem of global optimizing especially in identifying a better approximating value to that of the global minimum of any provided function in a particular search space that is large. This is in analogy with the annealing process having a formal proof model for faster convergence. The SA and its behavior are kept under control with the help of cooling scheme which doesn't depend on the first solution. Due to certain problems, Simulated Annealing is much efficient when compared to exhausting list as the objective to obtain a solution that is acceptable within a reasonable time. The SA algorithm has some demerits like needing many iterations for generating near optimal or optimal solutions and the initial temperature is hard to determine.

The data mining part of the P2P network has been dealt with in this work. In this work, the k-means clustering is duly optimized by the multi-objective fish swarm optimization along with simulated annealing algorithm that uses neighbour cooperation in the peer to peer network. In Sect. 2 some related literature is given. The techniques and methods used are explained in Sect. 3 and the outcomes and suggestions are given in Sect. 4 which Sect. 5 completes this work.

2 Literature survey

The MultiObjective PSO (MOPSO) which was set-up during 1999 is now an emerging field to solve the MOOs having extensive literature, applications, codes, variants, and software. Lalwani et al. [8] reviewed the applications of the MOPSO in different areas and studied the MOPSO variants as well. The MOO and its key concepts were studied and reviewed in the survey that was organised with multiple objectives and their variants.

Singh et al. [4] brought about a Simulated Annealing protocol in the conditioned multi-objective optimization (MOO). While traversing for it in a possible area, this protocol performs as it is in the current put-forth archived multi-objective simulated annealing (AMOSA) algorithm, and while working in an area that is not feasible it brings down the violation constraint by means of moving through the approximate descent direction (ADD). The Archive of the non-dominating results that are determined at the time of search was kept. The possibility of acceptance of one new point has been found by the status of its possibility and the status of dominating after comparing with the current points as well as the archives point. The proposed algorithm's performance has been reported with a group of constrained

bi-objective test problems (CTP2 to CTP8) seven in number that poses difficulties to the current protocols. A comparative analysis of this protocol used in wide range evolutionary algorithms that are multi-objective called the NSGA-II is included.

Jiang and Zhu [9] made a proposal for an approach that uses AFSA for solving the multi-objective problem of optimization. In this, the Pareto dominance concept was used for evaluating the advantages like faster convergence rate, ability to search globally, robustness and disadvantages of the Artificial Fish (AF). The Artificial fish search uses the technique of parallel search in the available solution space. The Pareto optimal solutions found is saved in an External Record Set. The effectiveness of the algorithm was illustrated by the results obtained through the simulation of 4 benchmark test functions.

Jiang and Cheng [10] made a presentation of a stochastic approach known as simulated annealing-artificial fish swarm algorithm (SA-AFSA) to find solutions to multimodal issues. This algorithm incorporates both the AFSA and the SA for improving performance. This hybrid algorithm has some features like strong local search abilities to solve few multimodal problems. This protocol uses the Simulated Annealing to the AFSA so that AFSA's performance is improved. This hybrid protocol possesses the following-characteristics such as the hybrid algorithm maintains a strong local searching capability of the Simulated Annealing along with swarm intelligence of the AFSA. The experimental outcomes show that as in the other testing case the SA-AFSA will get good precision and converging speed.

Many multi-objective algorithms for automatic clustering are put-forth in literature to handle the clustering problem. Abubaker et al. [11] made a proposal of a Multi-Objective Particle Swarm Optimization along with a simulated annealing algorithm (MOPSOSA). The MOPSOSA aims at the estimation of the right number of clusters and groups them into a data set which need not identify the actual cluster number. Its efficiency was studied on the basis of the velocity of the particles. Certain synthetic along with real life datasets were employed to measure the efficiency of this MOPSOSA algorithm. The results proved that some suitable velocity parameters belonging to the same range have been used.

Fang et al. [12] further proposed another artificial fish swarm algorithm (MOAFSA), that imitated the behaviour of the fish in local search that makes use of a quick sort method for getting a solution set that was nondominated and which cuts the external set to that of the crowding distance. The paper used the MOAFSA for the functions test that was multi-objective. The results have proved that the MOAFSA has a higher speed of convergence and the Pareto set that corresponds to this is quite evenly distributed. The MOAFSA is now applied to schedule the optimization of the reservoir of a hydropower station.

The K-means clustering and their partitions are a collection of the tuples of data into a K disjoint, which are the exhaustive groups or clusters in which the K will be used in in a manner that is user specified. The main goal here is finding a clustering that brings down the sum of its distances among the data tuple and also the centroid of such clusters. The K-means will begin with one initial set of the K centroids that are selected randomly. The AFSA is that algorithm which has been based on the schools of simulation of the fish behavior. In the P2P system, in accordance to the flow of such service nodes and the provision of service nodes. For enjoying this the quality of service has been regarded as the food and its quality. The node which further enjoys this can be able to provide the service and therefore the node will also be the fish and the food. This node will send a process of service request that flows to the process of food. This quality of the service is regarded as the food and its quality. The node which enjoys such service will be able to provide the service and so the node may also be the fish and the food. Owing to the interaction process along with the artificial fish model with a high level of similarity that will be able to put this P2P model in a model of artificial fish swarm. The main disadvantages in this are the high level of time complexity, the lack of a balance among both the global and the local search and this is in addition to that of the lack of such benefitting in the experiences of the members in the group for the movements that are subsequent to this [13,14]. Here in this work the MOFSASA method that is proposed in a P2P system.

3 Methodology

The k-means clustering, the Artificial Fish Swarm Algorithm, the Simulated Annealing, the neighbour selection and the multi-objective FSA are discussed here.

3.1 Skin segmentation data set

Over B,G,R color space, the construction of the skin segmentation dataset (Table 1) is done. From the facial images of diversity, age, gender and race of people, the skin and the non-skin dataset are generated by using the skin textures.

Data set information From the facial images across various age groups including the young, middle and old, ethnicities like Caucasian, African, Asian etc., and genders that have been obtained from the FERET database and the PAL database, the B,G,R values that are obtained are randomly sampled for collecting the skin dataset. The sample size was totally 245,057. These constituted 194,198 non-skin samples and 50,859 skin samples.

Attribute Information: A dataset of dimension 245,057 * 4 is constituted here. The first 3 columns are B,G,R (x1, x2,

Table 1 Skin segmentation dataset

Dataset characteristics	Univariate
Attribute characteristics	Real
Associated tasks	Classification
Number of instances	245,057
Number of attributes	4
Missing values?	N/A
Area	Computer
Number of web hits	106,563

and x3 features) values and the 4th column is that of the class labels or the decision variable y

3.2 Optimized k-means clustering algorithm

The k-means protocol uses additional time for the calculation of the distance from the center of the cluster [15]. This additional time is saved by this method. In case the distance from a new cluster center is small than the distance of the object from the previous cluster center then recalculate the cluster center and evaluate the distance of the object from another center until there are no changes in the mean. This is done by the maintenance of two data structures used for storing the label of the cluster as well as the distance of the object that corresponds to the cluster's centre. The distance of this data object is updated for every iteration until the termination is reached. The optimized k-means algorithms' process is as below:

The algorithm that is optimized needs two data structure clusters for storing the cluster label and the distance for every iteration that is used for the iteration that comes next.

3.3 Multi-objective optimization based on artificial fish swarm algorithm (MOAFSA)

The AFSA's are superior to the other algorithms in terms of speed of convergence, global search capacity, and robustness. The AFSA that has global information gets applied in solving the problems of optimization. This algorithm has been initiated at a single node [16] which can be N1. This generates a set of centroids initially $V_1^{(1)} = \{\vec{v}_{j,1}^{(1)} : 1 \leq j \leq K\}$ using a termination threshold $\gamma > 0$ (that is a user-defined constant), sending them to their immediate neighbours $\Gamma^{(1)}$, and starts the first iteration. If a node gets the centroids and the γ for the very first time, the rest is sent to its immediate neighbours and the first iteration begins. After this, all nodes enter iterating mode 1 having similar initial centroids and the same terminating threshold γ. This k-means protocol under goes many iterations and it is modified at every node N_i. For every iterations

Input: The actual number of the desired clusters, k, and that of a dataset D = (d1,d2,.............,dn) which contain n data objects.
Output: A set of the k clusters.
Steps:
1) The k data objects are randomly selected from data-set D as its initial clusters.
2) The Euclidian distance d (di,cj) for every data object di (1 <= i<= n) with every cluster center cj (1 <= j<= k) is computed.
3) Every data object di, closest centre cj is found and assigned to the cluster center cj.
4) The label and distance is stored as:
Set the cluster[i] = j, and dist[i] = d (di, cj).
In which j is the cluster label where the data di resides and d (di, cj) the distance of the data object di to that of the cluster center that is labeled by j.
5) For every cluster j (1 <= j<= k) the new cluster center is recalculated.
6) For every data object di, the distance to the new center of the cluster is computed.
a) In case the distance is less or equal to the dist[i], the data object remains there; and the iteration stopped
b) Else
For each of the cluster center cj (1<=j<=k), the distance d (di, cj) is computed and assigned to the data object di with the closest cluster.
Set the cluster[i] = j;
Set the dist[i] = d (di, cj).
Repeat from step 3 until the center remains same.

the centroids and the counts of clusters l, N_i are collected duly from the immediate neighbour. This and the local da of N_i, are utilized for the generation of centroids for the next iteration. If they are substantially different then the N_i moves to the next iteration. Else it stays in a finished state. The flowchart of this AFSA algorithm is depicted in Fig. 1.

3.4 Simulated annealing (SA)

The SA [4] is a method of search that is point-to-point having a strong base. Its ability to use unfavorable solutions are used to prevent its convergence into local optima making it more dynamic. This SA was designed originally from optimization of single objectives because of its point-to-point nature of the search. Certain earlier attempts were made in combining many objectives into one single composite function with weights.

Simulated annealing [17] is an algorithm that is straightforward where the Metropolis Monte Carlo technique is used. This is much suitable for the Simulated Annealing as the states that are possible only are sampled here at a particular temperature. This protocol is a Monte Carlo simulation run at very high temperature analogous to greater random fluctuation in modifying the variables. Later it is carefully monitored to make sure that the search space converges to the optimal level during the simulation as the temperature is reduced analogous to have less fluctuation in modifying the variables. Simulated annealing (SA)

Fig. 1 Flowchart of proposed
AFSA algorithm
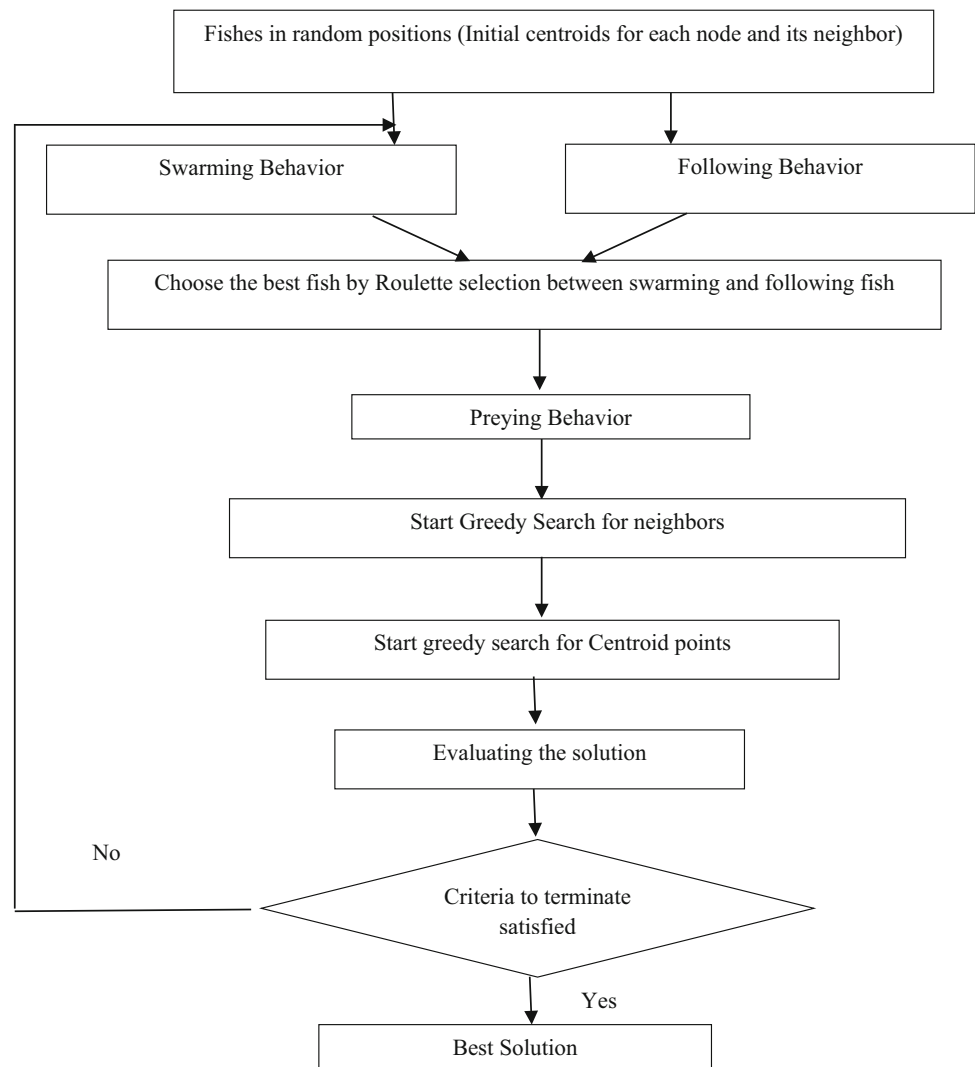
is also used to search the optimal solutions of problems using determination of initial (high) as well as final (low) temperature analogous kT (in which k refers to a Boltzmann's constant) in case of the Acceptance Checking, as well as in finding which will constitute a Monte carlo step.

The initial, as well as the final temperature in a problem, is found using the acceptance of probability. Generally, if Monte Carlo simulating permits energy (E) an increment in dEi at probability of Pi, the effective initial temperature is $kTi = -dEi/ln(Pi)$. When during last temperature there is a cost increment of 10 will be welcomed using a probability of about 0.05 (5%), the last temperature being $kTf = -10/ln(0.05) = 3.338$

Algorithm
Start the system with a known configuration, at known energy E
T = temperature = hot; frozen = false;
While (! frozen) {
repeat {
Perturb system slightly (e.g., moves a particle)
Compute E, change in energy due to perturbation
If($\Delta E < 0$)
Then accept this perturbation, this is the new system config
Else may be accepted, with probability = exp($-\Delta E/KT$)
} until (the system is in thermal equilibrium at this T)
If(ΔE still decreasing over the last few temperatures)
Then T=0.9T//cool the temperature; do more perturbations
Else frozen=true
}
return (final configuration as low-energy solution)

3.5 Fish swarm algorithm-simulated annealing (FSASA)

The noted swallow behaviour of the swarming fish has been presented in SA-AFSA for reducing its complex nature. Its ability to search globally for the AFSA relates to AFs population. Once iterations are complete, if the fitness function of the AF is bad compared to a provided threshold there is small control over the optimum outcome. The weak fish will take more computing time, the storage and thereby increases the toughness of the algorithm [18]. The behaviour of swallowing that is explained here will solve this issue and the AF that is weak will be swallowed if the value of the fitness function value is below the mentioned threshold value.

The implementation of SA-AFSA is as shown for global minimum, for random fish using Random (0, 1) to generate random number between 0 and 1. The variable *af_total* defines the number of AF, the *AFS_iterate_time* shows its iterations in the AFSA, *af_step* depicts a step distance of the AF, *af_visual* is the visual distance of the AF, *af_delta is the* crowd factor, *try_number* is the maximum try number of the AF, *T0 is* the initial temperature, *c_rate is* the rate of cooling, *SA_iterate_time_max* is maximum iterations in the SA, *k*denotes the number of iteration taking place at a given temperature.

The algorithm for AFSA is explained below

(a) Initialization of the parameters: *af_total*, *AFS_iterate_time*, *af_step*, *af_visual*, *af_delta*, *try_number*, *T*0, *c_rate*, and *SA_iterate_time*.

(b) Create the *af_total* AFs within the area of feasible solutions.

(c) Update the locations of each artificial fish. The locations of the AF's are updated dynamically as below: Execute its swallowing behavior and on the condition being satisfied in (1):

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{||X_j - X_i^t||}.af\_step.Rand() \tag{1}$$

X_j denote the fish which is to be followed

In case the condition of the swarming behavior is satisfied go forward to follow behavior else go to prey behavior in (2):

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{||X_c - X_i^t||}.af\_step.Rand() \tag{2}$$

X_c denotes the center position of the strong fish and its swarm's position.

Default:

Its preying behavior is executed, the AF chooses a state X_j stays randomly within its visual distance in (3).

$$X_j = X_i + Rand().af\_visual \tag{3}$$

If X_j is better than the current state, it moves a random step forward in the direction in (4).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{||X_j - X_i^t||}.af\_step.Rand() \tag{4}$$

In case a better location is not identified after many trials the AF makes a random step.

(d) Step **b** is continued till the condition of termination is met and the best solution *best X* is recorded and found by the AFSA.

(e) when i $< SA\_iterate\_time\_max$ do:

(1) for j = 1:k
(2) generating another new solution X_{new} using X_{best}, and compute its fitness solution, and keep a track of any variation ΔF that exists between X_{new} and X_{best} in (5):

$$\Delta F = f(X_{new}) - f(X_{best}) \tag{5}$$

(3) if $\Delta F < 0$, $X_{best} = X_{new}$
(4) if $\Delta F > 0$, p= $\exp(-\Delta E/T(i))$, if p > Random(0,1), *best new X = X*, else its *best X* remains unchanged.
(5) end

(f) increase the times of iteration i = i+1; and by cooling of the systems temperature T(i) = T0 * c_rate;

(g) Print the solution-vector as its final solution.

3.6 Multi-objective fish swarm algorithm simulated annealing (MO-FSASA) algorithm

The MO-FSASA algorithm [11] hereby integrates all the benefits of quick computation as well as converging in PSO which has the capacity to evade local solutions of Simulated Annealing. This MO-FSASA protocol begins with k-means scheme for creating initial positions of the swarm. After this, the Multi-objective Fish Swarm Optimization (MFSA) is made in which the swarm particles are introduced to search space by using the current optimal particles so that the best solution is identified. When searching, the Multi-Objective Simulated Annealing (MOSA) is made use in case if the position of the particle doesn't change and it does not progress to a preferred position. Every iteration makes use the concept to share an update of fitness in the repository of the solutions that are Pareto optimal. In Fig. 2 a conceptual diagram of the MOFSASA algorithm and its mechanism along with the input and the output datasets are shown. This algorithm has automatic clustering in accordance to different index that validate clusters which are the DaviesBouldin index (DB-index), based on the Euclidean distance; symmetry-based index (Sym-index), based on the

point symmetry distance; connectivity-based index (Conn-index), depending on the shorter distance. The mapping is carried out by [19].

The MO-FSASA algorithm is explained

Step 1: Initializing the swarm size n, iteration number $iter, k_{min}, k_{max}, T_0$ and t = 0.

Step 2: Using k-means technique for generating initial X_i^t, $V_i^t = 0$ and $XP_i^t = x_i^t$, i=1,....,n

Step 3: Computing $f_1(X_i^t), ..., f_s(X_i^t)$ repository of the non-dominated $XP_i^t, i = 1, ..., n$

Step 4: Selection of leader XG_i^t from repositories. Renumber XP_i^t and XG_i^t on $X_i^t, i = 1, ..., n$

Step 5: Computing X_i^{new} and V_i^{new}, i = 1,......,n.

Step 6: The validity of the X_i^{new}, i = 1,......,n is now analysed, and the position's validity is applied when the case is in-valid.

Step 7: Compute the $f_1(X_i^{new}), ..., f_s(X_i^{new})$ for the next position of the candidate X_i^{new} and $f_1(X_i^t), ..., f_s(X_i^t)$ for that of the current position $X_i^t, i = 1, ..., n$,

Step 8: Implementing a dominating check for X_i^{new}, i = 1; : : : ; n. If X_i^{new} not dominated by X_i^t then $X_i^{t+1} = X_i^{new}$ and $V_i^{t+1} = V_i^{new}$. Else $X_i^{t+1} = X_i^{MOSA}$ and $V_i^{t+1} = V_i^{MOSA}, i = 1, ..., n$ in which X_i^{MOSA} is the position, V_i^{MOSA} the velocity being returned from MOSA technique.

Step 9: Identify a new $XP_i^{t+1}, i = 1, ..., n$.

Step 10: Updating the repository.

Step 11: Set t = t + 1, if t > iter terminate the process of the algorithm, and this repository has the Pareto solutions, else traverse back to step 4.

3.7 Collaborative distributed fuzzy C-means clustering (CDFCM) algorithm

The novel objective function developed as (6) is the CDFCM algorithm under the P2P network environment. This is done by combining the distributed weight dissimilarity measure and an extra term for the attribute-weight-entropy regularization.

$$\min F(U, C, M) = \sum_{j=1}^{J} \sum_{n=1}^{N_j} \sum_{k=1}^{K} u_{jnk}^{\alpha}$$
$$\sum_{m=1}^{M} w_{jkm}(x_{jnm} - c_{jkm})^2$$
$$+ \gamma \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} w_{jkm} \log w_{jkm}$$
$$Subject\, to\, c_{jkm} = c_{ikm}, i \in NB_j$$
$$\sum_{k=1}^{K} u_{jnk} = 1, 0 \le u_{jnk} \le 1$$
$$w_{jkm} = w_{jkm}, i \in NB_j$$
$$\sum_{m=1}^{M} w_{jkm} = 1, 0 \le w_{jkm} \le 1 \qquad (6)$$

where U = $[u_{jnk}]$ is the membership degree matrix and u_{jnk} denotes the membership degree of the n-th object belonging to the k-th cluster in the j-th peer. C = $[c_{jkm}]$ is the cluster prototype matrix and c_{jkm} denotes the m-th dimension of the k-th cluster prototype in the j-th peer. W = $[w_{jkm}]$ is the attribute weight matrix and w_{jkm} denotes the m-th dimension of the k-th cluster weight vector in the j-th peer. α is the fuzzification coefficient and γ is a positive scalar.

The shape and the size of the clusters is controlled by the first distributed weighted distance term in this new objective function [20]. This also facilitates the agglomeration of clusters. The negative entropy of the attribute weights is constituted in the second term. The optimal distribution of all the attributes are regularized by the attribute weights, in accordance with the data which is available. This is done in order to minimize the intra-cluster dispersion simultaneously maximizing the entropy of the attribute weights for stimulating significant attributes which can contribute better for the cluster identification . A positive regularizing and an adjustable parameter is γ ($\gamma > 0$), whose proper selection enables the balancing of the two terms so that an optimal solution can be found. The local cluster prototypes are ensured by consensus constraints $c_{jkm} = c_{ikm}$ and, $w_{jkm} = w_{ikm}$ i$\in$ NB$_j$. along with yielded attribute weights at each peer coinciding with the global ones of all the objects. i.e. the results obtained by centralized clustering technique are similar to the results obtained by distributed clustering.

Minimizing F (U, C, W) with respect to constraints is a constrained nonlinear optimization problem. This problem is solved by applying the Picard iteration, similar to the conventional FCM algorithm.After initially fixing C and W, it finds the required conditions on U for reducing F(U) to the least value and then W and U are fixed to decrease F(C) to the least value with regard to C. The last step is the fixing of U and C so that F(W) is minimized with respect to W.

Fig. 2 Framework for MO-FSASA algorithm
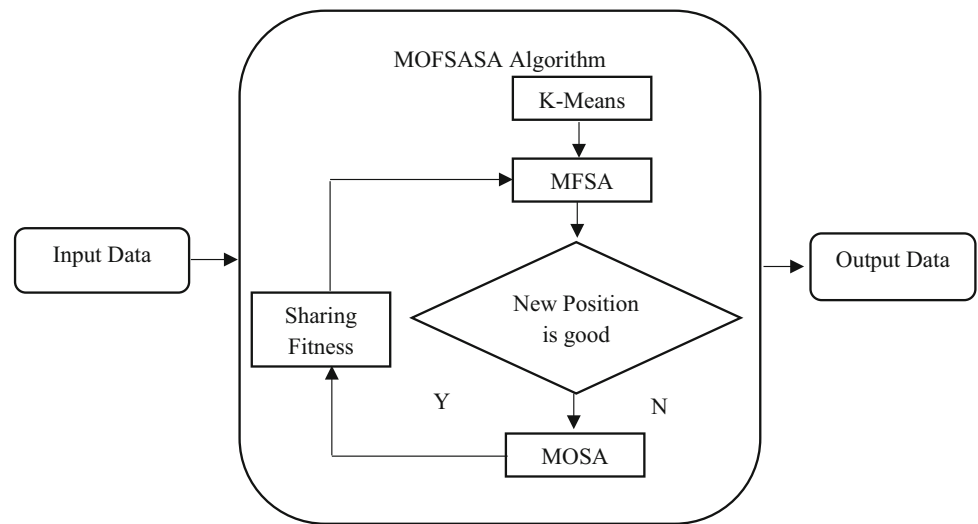
Corresponding to equations (7) through (11), the matrices U, C and W are respectively updated.

$$u_{jnk} = \frac{1}{\sum_{h=1}^{K} \left(\frac{\sum_{m=1}^{M} w_{jkm}(x_{jnm}-c_{jkm})^2}{\sum_{m=1}^{M} w_{jhm}(x_{jnm}-c_{jhm})^2} \right)^{\frac{1}{\alpha-1}}}$$

$for\ 1 \leq j \leq J, 1 \leq n \leq N_j, 1 \leq k \leq K$ (7)

$$c_{jkm} = \frac{\sum_{n=1}^{N_j} u_{jnk}^{\alpha} w_{jkm} x_{jnm} - \sum_{i \in NB_j} p_{jikm}}{\sum_{n=1}^{N_j} u_{jnk}^{\alpha} w_{jkm}}$$

$for\ 1 \leq j \leq J, 1 \leq k \leq K, 1 \leq m \leq M$ (8)

$$p_{jikm} = p_{jikm} + \eta_1(c_{jkm} - c_{ikm})$$

$for\ 1 \leq j \leq J, 1 \leq k \leq K, 1 \leq m \leq M, i \in NB_j$ (9)

$$w_{jkm} = \frac{\exp\left(-\gamma^{-1} \sum_{n=1}^{N_j} u_{jnk}^{\alpha}(x_{jnm}-c_{jkm})^2 - 2\gamma^{-1} \sum_{i \in NB_j} q_{jikm}\right)}{\sum_{l=1}^{M} \exp\left(-\gamma^{-1} \sum_{n=1}^{N_j} u_{jnk}^{\alpha}(x_{jnl}-c_{jkl})^2 - 2\gamma^{-1} \sum_{i \in NB_j} q_{jikl}\right)}$$

$for\ 1 \leq j \leq J, 1 \leq k \leq K, 1 \leq m \leq M$ (10)

$$q_{jikm} = q_{jikm} + \eta_2(w_{jkm} - w_{ikm})$$

$for\ 1 \leq j \leq J, 1 \leq k \leq K, 1 \leq m \leq M, i \in NB_j$ (11)

Here P = $[p_{jikm}]$ and Q = $[q_{jikm}]$ are two matrices containing the Lagrange multipliers corresponding to the consensus constraints $c_{jkm} = c_{ikm}$ and, $w_{jkm} = w_{ikm}$ i∈ NB_j. They are defined for the iterative update of the cluster prototypes and attribute weights. η_1 and η_2 are positive scalars.

Exploring the structures of every peer through peer exchanges is the essence of collaborative clustering. Clustering at individual peer and an interaction between neighbor peers by means of exchanging the findings are the two main phases of collaborative clustering. These intertwine and take place at fixed sequence. Figure 3 shows a general view of the processing of the proposed collaborative distributed clustering algorithm.

In the beginning, every peer spawns its initial cluster prototypes and attribute weights. The local findings are then
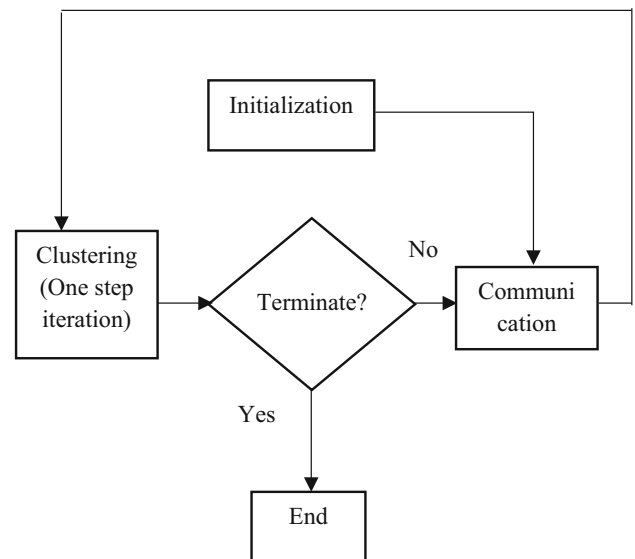
Fig. 3 The block diagram of the proposed collaborative distributed clustering algorithm

conveyed to its neighbors. Then, at every peer, the FCM-type algorithm is performed with its optimization pursuits. This is carried out by focusing on local data and at this point in time, the findings are communicated by the neighbor peers. All of the peers are set to involve in another communication phase after one step iteration. Once more, the findings are communicated by them and novel conditions for the new phase of the FCM-type clustering are set up. Collaboration refers to the pair of the processes of communication and clustering. After taking a finite number of collaboration iterations, the overall optimization stops when there is no more considerable improvisation in the exposed structure (cluster prototypes and attribute weights) of all the peers.

CDFCM Clustering Algorithm

Input: The number of peers J, the number of objects generated by the j-th peer N_j, the number of data dimensions M, the number of clusters K, parameter α, γ, $η_1$ and $η_2$.

Output: Accuracy of the clustering.

For each peer j (j ∈ {1, 2, ..., J}), randomly generate K initial cluster prototypes C (0), set the value of initial attribute weight to 1/M, set p_{jikm} (0) = 0 and q_{jikm} (0) = 0 for i∈ NB_j ,1 ≤ k ≤ K, 1 ≤ m ≤ M, broadcast its initial cluster prototypes and attribute weights to its neighboring peers, set iteration index t=0;

Repeat

D: Update the partition matrix U (t+1) by (7) at each peer;

D: Update the cluster prototypes C (t+1) by (8) at each peer;

Each peer broadcasts the updated cluster prototypes to all its neighboring peers;

D: Update the multipliers P (t+1) by (9) at each peer;

D: Update the attribute weights W (t+1) by (10) at each peer;

Each peer broadcasts the updated attribute weights to all its neighboring peers;

D: Update the multipliers Q (t+1) by (11) at each peer;

t++;

Until termination condition of the collaboration activities has been satisfied.

The algorithm involves distributive mode of processing the iterative steps with "D" marks. Based on the local findings of every peer and those coming from the neighbors, the cluster memberships, cluster prototypes and attribute weights are correspondingly updated by every peer. The peer sends "convergence" message to its neighbors when the variation of the cluster prototypes and the attribute weights in two successive iterations is lesser than the preset threshold. When a peer receives the "convergence" notification from all its neighbors, the algorithm iteration of each peer terminates. Then, is obtaining the consensus on cluster prototypes and attribute weights. Due to the fact that the clustering performance coincides with that of the centralized clustering techniques, the data transmission energy is reduced and evened out among the peers by the CDFCM algorithm.

4 Results and discussion

This work makes use of the MATLAB simulation environment for evaluating the proposed techniques.

Table 2 Parameters for AFSA

Parameter	Value
Number of population	15
Visual	9
Step	1
Maximum try number	5
Crowd factor δ	0.8
Distance D	500
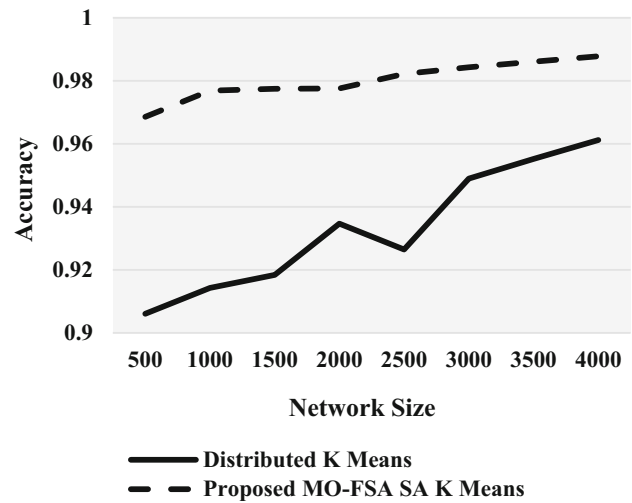
Fig. 4 Accuracy for proposed MO-FSASA K means compared with distributed K means

Table 2 shows the parameters for AFSA. Figures 4, 5, 6 and 7 shows the accuracy and F Measure for Proposed MO-FSASA K Means respectively. The Fig. 8 shows the fitness and Fig. 9 shows the standard deviation.

Accuracy considers both true positives (TP) and true negatives (TN) over all instances. In other words, accuracy shows the ratio of all correctly classified instances in (12).

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

F-measure is a harmonic mean of precision and recall in (13, 14, 15).

$$\frac{2 \times (\text{Pr}ecision \times Recall)}{\text{Pr}ecision + Recall} \tag{13}$$

where

$$\text{Pr}ecision = \frac{TP}{TP + FP} \tag{14}$$

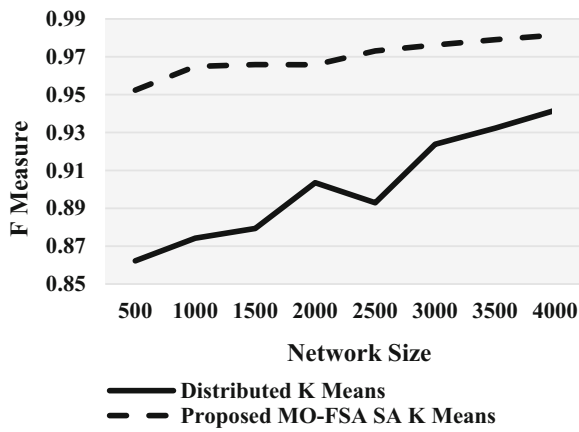$$Recall = \frac{TP}{TP + FN} \tag{15}$$

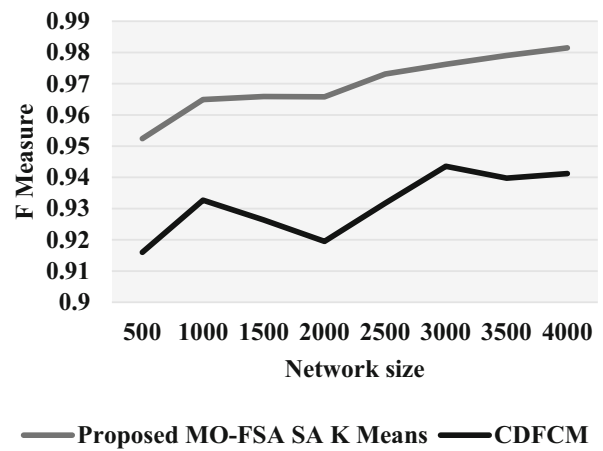**Fig. 5** F measure for proposed MO-FSASA K means compared with distributed K means

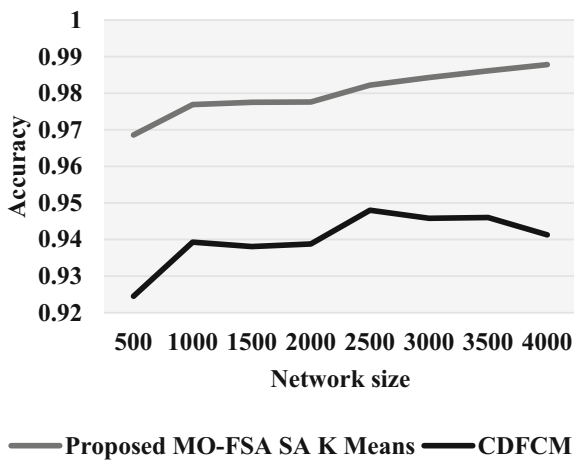**Fig. 6** Accuracy for proposed MO-FSASA K means compared with CDFCM

Fig. 7 F measure for proposed MO-FSASA K means compared with CDFCM

Fig. 8 Fitness

Fig. 9 Standard deviation

It is observed from Fig. 4 that the accuracy of Proposed MO-FSASA K Means performs better by 6.67%, by 4.49%, by 3.65%, and by 2.73% for the network size 500, 2000, 3000 and 4000 respectively than distributed K Means Clustering.

It is observed from Fig. 5 that the F Measure of Proposed MO-FSASA K Means performs better by 9.94%, by 6.67%, by 5.52%, and by 4.13% for the network size 500, 2000, 3000 and 4000 respectively than distributed K Means Clustering.

From the Fig. 6, it can be observed that the proposed MO-FSA SA K Means has higher classification accuracy by 4.65% for 500 number of size, by 4.04% for 2000 number of size, by 3.98% for 3000 number of size and by 4.82% for 4000 number of size when compared with CDFCM.

From the Fig. 7, it can be observed that the proposed MO-FSA SA K Means has higher f measure by 3.89% for 500 number of size, by 4.91% for 2000 number of size, by 3.4% for 3000 number of size and by 4.18% for 4000 number of size when compared with CDFCM.
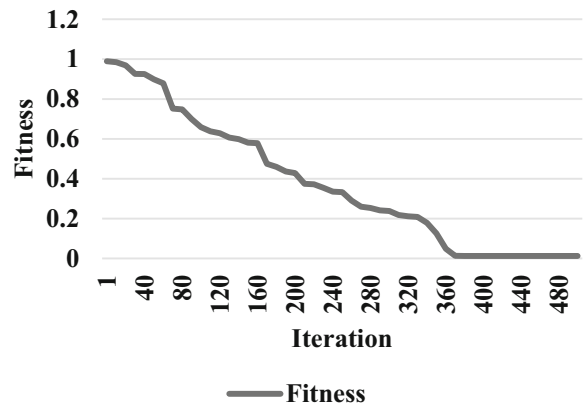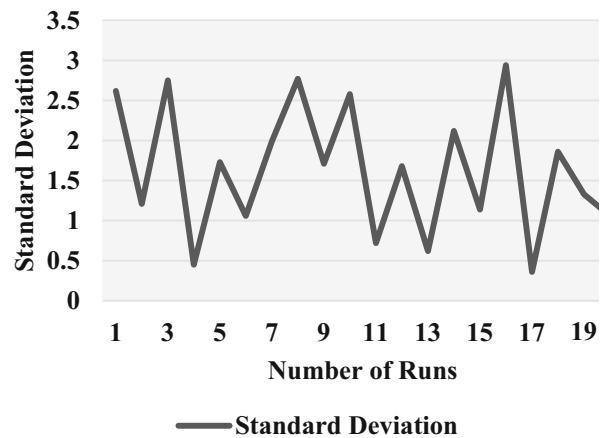
From the Fig. 8, it can be observed that the fitness function achieves better performance through number of iterations.

From the Fig. 9, it can be observed that the standard deviation increased through number of runs.

5 Conclusion

The peer-to-peer networking paradigm such as decentralization, self organizing etc. helps in building the network applications. MO-FSASA protocol was used for improving the neighbour cooperation in the peer-to-peer network application to speed up the file transfer, share redundant computing power for enhancing the performance. SA hybridized with AFSO algorithm based on the swallowed behaviour greatly improved the stability and the global search ability thus speeding the convergence. This shows more benefits like a high speed of convergence, high accuracy, fault tolerance, and flexibility. The main aim of the MOFSASA will be to estimate a suitable cluster number to partition the data set within such clusters with no need to know the clusters. The ability in obtaining the accurate solutions will be the most critical objective in the process of clustering.

The results have shown that this proposed MO-FSASA K Means performs with a better accuracy by about 6.67, 4.49, 3.65 and 2.73% for network size 500, 2000, 3000 and 4000 respectively than that of the distributed K Means Clustering. Likewise, the F Measure of the Proposed MO-FSASA K Means performs with a better accuracy by about 9.94, 6.67, 5.52 and 4.13% for network sizes of 500, 2000, 3000 and 4000 respectively than that of the distributed K Means Clustering.

The case of the future work the focus will be mainly to influence the proposed algorithm's influence and also the MOFSASA application in the other fields in the industry for the prediction of stock, the optimization of weights and the recognition of patterns. In case of the future work for the purpose of implementing the other algorithms that are meta-heuristic.

References

1. Brienza, S., Cebeci, S.E., Masoumzadeh, S.S., Hlavacs, H., Özkasap, Ö., Anastasi, G.: A survey on energy efficiency in P2P systems: file distribution, content streaming, and epidemics. ACM Comput. Surv. (CSUR) **48**(3), 36 (2016)
2. Said, G.A.E.N.A., Mahmoud, A.M., El-Horbaty, E.S.M.: A comparative study of meta-heuristic algorithms for solving quadratic assignment problem. arXiv preprint arXiv:1407.4863 (2014)
3. Forero, P.A., Cano, A., Giannakis, G.B.: Distributed clustering using wireless sensor networks. IEEE J. Sel. Top. Signal Process. **5**(4), 707–724 (2011)
4. Singh, H.K., Isaacs, A., Ray, T., Smith, W.: A simulated annealing algorithm for constrained multi-objective optimization. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 1655–1662. IEEE (2008, June)
5. Datta, S., Giannella, C., Kargupta, H.: Approximate distributed k-means clustering over a peer-to-peer network. IEEE Trans. Knowl. Data Eng. **21**(10), 1372–1388 (2009)
6. Yang, X.S.: Nature-Inspired Optimization Algorithms. Elsevier, Amsterdam (2014)
7. Liu, Y., Xiong, S., Liu, H.: Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP. In: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 895–898. ACM (2009, June)
8. Lalwani, S., Singhal, S., Kumar, R., Gupta, N.: A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm. Trans. Comb. **2**(1), 39–101 (2013)
9. Jiang, M., Zhu, K.: Multiobjective optimization by artificial fish swarm algorithm. In: 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 3, pp. 506–511. IEEE (2011, June)
10. Jiang, M., Cheng, Y.: Simulated annealing artificial fish swarm algorithm. In: 2010 8th World Congress on Intelligent Control and Automation (WCICA), pp. 1590–1593. IEEE (2010, July)
11. Abubaker, A., Baharum, A., Alrefaei, M.: Multi-objective particle swarm optimization and simulated annealing in practice. Appl. Math. Sci. **10**(42), 2087–2103 (2016)
12. Fang, G., Guo, W., Huang, X., Si, X., Yang, F., Luo, Q., Yan, K.: A new multi-objective optimization algorithm: MOAFSA and its application. Przeglad Elektrotechniczny **88**(9b), 172–176 (2012)
13. Liu, J., Liu, X., Zhu, C.Y., Bai, Y.Y.: A P2P reputation incentive mechanism based on artificial Fish-Swarm model. In: Applied Mechanics and Materials, vol. 321, pp. 2725–2731. Trans Tech Publications (2013)
14. Neshat, M., Sepidnam, G., Sargolzaei, M., Toosi, A.N.: Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. Artif. Intell. Rev. (2014). https://doi.org/10.1007/s10462-012-9342-2
15. Tiwari, S., Solanki, T.: An optimized approach for k-means clustering. Int. J. Comput. Appl. (0975–8887). 9th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine-2013), pp 5–7
16. Saha, S., Bandyopadhyay, S.: Some connectivity based cluster validity indices. Appl. Soft Comput. **12**, 1555–1565 (2012). https://doi.org/10.1016/j.asoc.2011.12.013
17. Kumar, M.: A comparative study using simulated annealing and fast output sampling feedback technique based PSS design for single machine infinite bus system modeling. Int. J. Eng. Res. Appl. (IJERA) **2**(2), 223–228 (2012)
18. Mingyan, J., Yongming, C., Dongfeng Y.: Improved artificial fish swarm algorithm. In: Proceedings of the 5th International Conference on Natural Computation (ICNC'09), 14–16 August, Tianjin, China (2009)
19. Baskaran, M., Sadagopan, C.: Synchronous firefly algorithm for cluster head selection in WSN. Sci. World J. **2015**, 7 (2015)
20. Zhou, J., Chen, C.P., Chen, L., Li, H.X.: A collaborative fuzzy clustering algorithm in distributed network environments. IEEE Trans. Fuzzy Syst. **22**(6), 1443–1456 (2014)

S. Raju is a Ph.D research scholar of Anna University-Chennai. He has received M.E.(Computer Science and Engineering) degree in 2008 from the Anna University-Chennai, Tamilnadu, India. His research interests includes Distributed Data Mining, Mobile Networks, Peer-to-Peer Networks. Currently he is working as a Ass. Professor and Head in Information Technology at Mahendra Engineering College, Namakkal, India and his experience in teaching started from the year 1995. He has received Best Teacher award. He has organized and participated more than 35 programs like Workshop, Seminar, Conference and Training. He is a faculty member of Computer Society of India and Life member of ISTE.

M. Chandrasekaran received his B.E.(Hons.) degree in Electronics and Communication Engineering from University of Madras and his M.E. degree in Computer Science and Engineering from Bharathiyar University. He completed his Ph.D. in Information and Communication Engineering under Anna University and his area of research is Computer Networks. He has more than 30 years of teaching experience. He has published more than 70 papers in National and International Conferences conducted by IEEE and more than 40 papers in National and International peer reviewed journals. He is a member of ISTE, CSI, IAENG, ACM, CSTA etc. His research interests include Neural Networks, Fuzzy Logic, Congestion control in TCP Networks and Sensor Networks. He served as member of various committees representing AICTE and DOTE, Tamil Nadu for inspecting Engineering Colleges. He served as a member of academic council. He guided many B.E., M.E. and M.C.A Projects. Some of the Projects got the Award from Tamil Nadu State Council for Science and Technology. He is currently working as Principal at Government College of Engineering, Bargur.