Electr Eng (2010) 92:269–282
DOI 10.1007/s00202-010-0178-y

ORIGINAL PAPER

# Clustering of voltage control areas in power system using shuffled frog-leaping algorithm

**F. Rameshkhah · M. Abedi · S. H. Hosseinian**

Received: 9 September 2009 / Accepted: 28 September 2010 / Published online: 27 October 2010


**Abstract** Determination of critical voltage control areas (VCAs) is a very important task in both voltage stability assessment and control. However, it is impossible to detect VCAs in real time during appearance of emergency cases for large-scale power systems. Therefore, it is a reasonable solution to employ an artificial intelligent system (AIS) to detect VCAs and to identify the prone buses for monitoring and control purposes as quickly as possible. The training data must contain the simulation results and the historical data collected from a wide range of various emergency cases. Using this database, a clustering process which provides finite clusters of all possible VCAs and a classification function which affiliates each emergency or stress case to its own cluster of VCAs are the main stages to prepare AIS for automatic VCA identification. In this paper a novel data clustering method based on shuffled frog-leaping algorithm (SFLA) is presented for the first task. The results are finite numbers of clustered groups of VCAs with a representative vector of participation factors (PF) for each group. SFLA combines the benefits of the genetic-based memetic algorithms as well as social behavior-based particle swarm optimization methods. In present study the application of SFLA in data clustering is also compared with the most popular analytic algorithm of clustering, *K*-means, and also with genetic algorithm-based data clustering to demonstrate the validity of proposed clustering method. Numerical results are also presented for IEEE 14-bus test system and an artificial database. The comparative results show the effectiveness of proposed algorithm.

F. Rameshkhah · M. Abedi · S. H. Hosseinian (✉)
Department of Electrical Engineering,
Amirkabir University of Technology, Tehran, Iran
e-mail: Hosseinian@aut.ac.ir

**Keywords** Voltage Control Area (VCA) · Participation factors (PF) · Data Clustering · Evolutionary algorithms (EV) · Memetic algorithms · Shuffled frog-leaping algorithm (SFLA)

## 1 Introduction

In recent years, large-scale power systems operate much closer to their limits. One of the major problems associated with such a stressed system is voltage instability. In fact, voltage stability has become the main technical factor which limits the power transfer capacity of transmission system in many electricity markets.

The two lines of defense against voltage instability and consequence blackout could be expressed as (a) Assessment and maintenance of adequate security margins against credible events and (b) System Protection Schemes (SPS) against more sever scenarios. In both of these situations, serious condition must be distinguished by security assessment. Consequently, decision related to how to act against the ongoing risk should be made faster than real time. However, accomplishment of these tasks for a large-scale power system even by advanced and high-speed computers is impossible in real-time environment. Historically, the first solution for this problem was designing and installing of the primary version of SPSs [1]. Afterward, using the various methods based on energy functions, miscellaneous creative rough indices for security assessment and contingency screening and ranking have been tested. In addition, remarkable efforts to create more simplified and efficient simulation methods such as Quasi Steady-State simulation (QSS) [2] are the other main issues that have been followed to cope with the referred requirements.

Recently, the application of Artificial Intelligent Systems as a suitable and practical solution for the aforementioned problem has attained increasing attention mainly due to the efficiency of present high-speed computers [3,4]. One of the main features, which can be attributed to (AIS)s such as Artificial Neural Networks (ANN), is their ability to learn nonlinear problem with selective training schemes in off line, leading to sufficient accurate online responses.

Definition of voltage control areas (VCA)s is one of the main objectives in the field of voltage stability and control. The (VCA)s are the areas in power systems that are prone to voltage instability under particular operating conditions and contingencies. In fact, all buses have specified participation factors in each case of voltage instability. The VCA is the set of all buses with higher participation factors in the current instability. Therefore, concentrating on control strategies for those prone buses of system instead of the whole buses not only provides these strategies to be more convenient for real-time simulation, but also forces the controllers to be more coordinated with each other and acts more efficiently. In addition, during the emergency and preventive control scenarios, it is more convenient to monitor the affects of control actions on electric power system, only by parameters of few buses (i.e. prone buses) instead of whole buses. These prone buses could be defined by VCA analysis. This approach can simplify monitoring and enhances the fastness of control actions during alert or emergency cases. There are well-known off line analytic methods to identify the VCA corresponding to each emergency or alert case of voltage instability [5,6]. However, for real-time decision- based control of emergency cases, it is essential to identify the corresponding VCA in real time too. Employing a trained ANN by using the historical data seems to be suitable for this purpose. On the other hand, for large-scale power systems with huge number of different voltage emergency cases, it is a complicated task and the effectiveness and accuracy of trained system would be poor. Fortunately, there are some data management contrivances that help to have more accurate results.

Although different contingencies and various stress levels and directions may provide different VCAs in power systems, it has been proved that many of those VCAs are very similar to each others. Therefore, clustering of various VCAs to limited number of groups is an effective approach to reduce the complication and size of artificial intelligent system (AIS) to enhance the reliability as well as simplification for training procedure of AIS.

Although several reports can be seen in the literatures about the application of intelligent systems in voltage stability assessment (VSA) and control, however, there are few reports available specifically about the use of intelligent systems in VCA identification. EPRI previously has published satisfactory results about the application of ANN in online voltage stability assessment for large-scale power systems. Recently, their research group has started to work on real-time identification of VCAs using the intelligent systems [7,8].

This paper is concentrated on VCA clustering, which is the first stage of the aforementioned on line VCA identification.

Data clustering is one of the main branches in the field of data mining. Several reports are available for data clustering procedure since the past tens years. However, in recent years, the clustering methodologies have been improved significantly due to developing new optimization techniques, especially evolutionary and memic-based algorithms,. The shuffled frog-leaping algorithm (SFLA) [9], is a meta-heuristic optimization method inspired from the memetic evolution of a group of frogs when seeking for food. The SFL algorithm, in essence, combines the benefits of the genetic-based memetic algorithms and the social behavior-based PSO algorithms. The SFLA have been used to solve discrete as well as continuous optimization problems. In [10], the authors proposed a hybrid multi-objective algorithm based on SFLA and bacteria optimization to solve a mixed model assembly line sequencing problem. Elbeltagi et al. [11] compared SFLA to other evolutionary-based algorithms such as genetic algorithms (GA), memetic algorithm (MA), particle swarm Optimization (PSO), and Ant Colony Optimization (ACO). In [12], an improved algorithm for SFLA implementation has been presented.

This paper aims to use of SFLA for clustering the data of bus participation factors resulted from the various situations of loading and contingencies which lead to emergency cases in power systems. The remainder of the paper is structured as follows: A brief description of VCAs, modal analysis and participation factors and also the concept of VCA clustering is presented in next section. Sect. 3 is an introduction to clustering methods. In Sect. 4, *K*-means algorithm as the most popular method for clustering is described. Sect. 5 is about the history and algorithm of SFL. In Sect. 6, the algorithm of data clustering based on SFLA has been described. The numerical results by applying SFLA-based data clustering on PFs data of IEEE 14-bus test system and also on an artificial test data in comparison of the *K*-means and genetic algorithms are presented in Sect. 7. A discussion about the properties of SFLA in data clustering is done in Sect. 8. In the last section we draw the conclusions.

## 2 Voltage control areas

Since it is well understood that voltage security is driven by the balance of reactive power in a system, it is of particular interest to find out at each case of imminent voltage instability what area in a system may suffer from reactive power deficiencies. The target of VCA determination is to correlate each contingency, load pattern and system condition which

leads the power system to an instability susceptible case, to its corresponding control buses in the system. Those control buses are the buses that implementing the control strategies on them is most effective to take the system far from the emergency states or to improve the loadability of the system. Since various combinations of contingencies and load patterns can lead to a huge number of emergency and alert cases, it is expected to have a wide variety of different VCAs in a large scale power system. However, simulation results show that in fact, the variety of the VCAs, even in very large power systems is very limited. In other words, the differences between many of those VCAs are limited to little differences of PF values of some of their members. Simply stated, the whole VCAs of a large-scaled system can be divided into a few clusters. The centers of mentioned groups (clusters) are good approximations for critical VCAs that are responsible for the emergency cases in the system. In this manner each contingency and load stress case would be attributed to one of this limited numbers of critical VCAs. Now a database including the monitored parameters of system at each simulated or experimented emergency cases of system (as input data) and the centriod of VCA clusters named critical VCAs (as output data) can be prepared. This collected database is used to train an ANN to attribute each arbitrary voltage emergency case to a critical VCA. This is in fact a tool for online identification of VCAs or a real-time classification of VCAs. The concept of VCA clustering which is the subject of this article, would be used to simplify the training of the ANN or any other intelligent systems to design a real-time VCA identifier. An almost similar approach has been mentioned in [7,8].

The identification procedure has three stages: (a) To determine (off-line) the specific buses and generators that form each VCA at each contingency or load stress case that leads the system to an emergency case for voltage (b) To cluster all determined VCAs to a proper number of groups and to specify the centriod of each group (c) To train an artificial system to attribute the mentioned contingencies and load stress cases to its related group of VCA.

The aim of this paper is to accomplish stages (a) and (b).

To determine VCAs in a given power system, the PV analysis for the base case under all probable load stresses and contingencies is performed. Results are checked against all security criteria such as reactive power reserve limitations. Stability is implied in convergence of the power flow. For cases that have no criteria violations, uniform increases in loading the system are continued out to the nose of the PV curve and the stability limits (the point in the transfer at which the nose occurs) are recorded. Modal (eigenvalue) analysis [6] is computed at the stability limit to determine the area of collapse as defined by the modal participation factors corresponding to the mode with a zero eigenvalue (bifurcation point). Based on these PFs, sets of buses and generators that form the various VCAs in a given power system are identified. The identification algorithm processes the sets of buses and generators corresponding to the PFs obtained from the modal analysis for each system condition and contingency case. As mentioned earlier, the algorithm divides the PF vectors of all of the generated cases to a finite number of clusters that each one has a representative VCA. Afterward, the algorithm classifies the set of all studied contingencies and load stress cases into classes corresponding to those clusters of VCAs. Contingencies and load conditions are among the same groups if their set of buses PFs are similar.

To provide the capability of real-time identification of VCA following an alert or emergency case, in power system, a sequence of off line stages must be done as follows:

1. Generate a wide variety of stress levels and directions (various loading levels), severe and more probable contingencies and combination of both aforementioned conditions for power system, using a proper simulation program.
2. Apply the off line analytic algorithm for determining the zero eigenvalue, its corresponding eigenvectors and finally the participation factors as in [5,6]. It is noticeable that distinguishing the cases which lead to emergency or alert states of voltage instabilities, among all above generated situations is in fact the duty of security assessment tools. However, in off line process for identification the VCAs, existing of a zero eigenvalue without or with uniformly increasing the load is the indicator of the proximity to voltage instability. In addition, some indicators such as maximum loadability factor and the magnitude of voltage can be used to specify the emergency and alert cases. However, if there is a proximity to voltage instability, regardless of being in emergency or alert case, the critical VCA is the set of prone buses to control the power system: Emergency control or preventive control.
3. Record the set of buses and generators that form VCA, their participation factors and the values of corresponding monitored parameters (such as voltage magnitudes, line currents, $P$ and $Q$ of generators and loads) at each of the emergency or alert cases in the database.
4. Cluster the whole determined VCAs in adequate number of groups (clusters).
5. Divide the database into training and test parts. Using the training data, attribute each contingency and system condition to its own VCA cluster and train an artificial intelligent system, to be able to do these tasks –on line- for each arbitrary contingency and/or load stress condition which occurs in system at real time. It means to attribute the power system situations at various contingencies and different system conditions to their own class of VCAs, or on line classification of con-

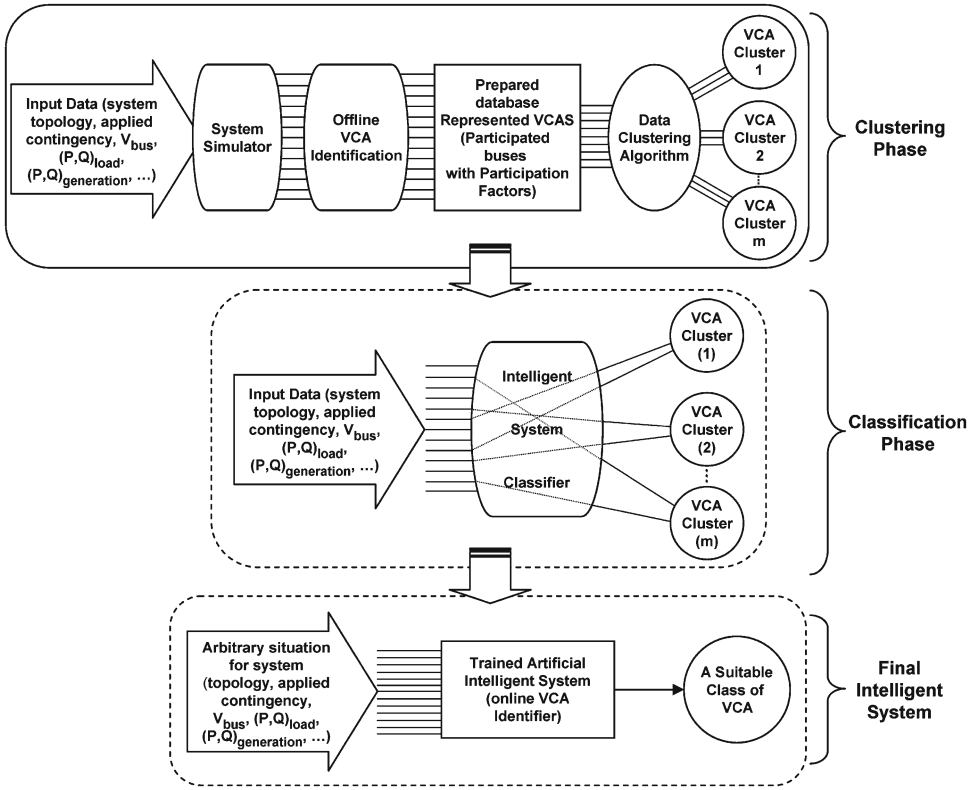


**Fig. 1** Schematic form of stages of building the final intelligent system

tingency and load stresses with respect to their critical VCA. It is noticeable that this article is concentrated on stages 1 to 4. Figure 1 illustrates this approach including a schematic form of stages of building the final intelligent system. This paper does not include the classification phase (the intelligent system) and as it was mentioned earlier the focus of this article is on the clustering phase.

## 3 Introduction to clustering

Clustering is one of the main tasks of knowledge discovery from databases (KDD) and one of the most widely studied topics in Data Mining. It has a wide range of applications in different branches of science and engineering. Clustering aims to discover sensible organization of objects in a given dataset by identifying and quantifying similarities (dissimilarities) between the objects. The fundamental clustering problem is to partition a given data set into groups (clusters), such that the data points in a cluster are more similar to each other than points in different clusters. In most clustering processes there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data. This is what distinguishes clustering from classification. In the other words, clustering is a mostly unsupervised procedure in contrast to classification which is a supervised process.

There are a multitude clustering methods available in the literatures, which can be broadly classified into the following types [13]: (a) partitional clustering; (b) hierarchical clustering; (c) density based clustering; and (d) grid-based clustering

Common clustering methods are based on the first two categories. A hierarchical clustering procedure achieves its clustering through a nested sequence of partitions, which can be represented in a treelike structure. On the other hand, partitional clustering method performs clustering in one shot. Historically, hierarchical clustering techniques have been more popular in biological, social and behavioral sciences, whereas partitional methods are more frequent in engineering applications. Table 1 shows a glance of advantages and disadvantages of the two categories of clustering methods. As it seems, the partitional methods usually lead to better results due to the nature of iterative and revised-type grouping method, and hence they are preferable if there is no emphasis on speed.

The most popular and well-known partitional clustering method is *K*-means clustering algorithm. It was invented in 1965 but it is interested yet because of its following advantages [13,14]:

**Table 1** Comparison of hierarchical and partitional methods of data clustering

| Hierarchical | Partitional |
|---|---|
| Proper speed | Relatively slow convergence |
| Number of clusters is determined automatically | Num. of clusters must be predefined |
| Probability to lead to incorrect results | Tend to produce better results |
| Related to the initial seed points for clusters | Related to the initial seed points for clusters |

(i) *K*-means algorithm converges extremely quickly in practice. In fact, many have observed that the number of iterations is typically much less than the number of points. However, it has been shown that there are certain point sets on which *K*-means takes super polynomial time.
(ii) Thanks to the quick convergence of *K*-means, the results would be used as a pre estimation of final clustering that is suitable to be considered as the initial values in advanced clustering algorithms like evolutionary based algorithms to make a more efficient hybrid clustering algorithm.
(iii) Though there is no guarantee of achieving a global minimum, at least the convergence of this algorithm is ensured [14].

On the other hand, *K*-means have some significant defects: In terms of performance, as mentioned earlier, the algorithm is not guaranteed to return a global optimum. As a classical iterative technique, the quality of the final solution depends largely on the initial starting conditions (initial centers of clusters), and may in practice, be much poorer than the global optimum.

However, through the extremely fast convergence of the algorithm, these defects would be moderated in many cases. For example, to overcome the lake of knowledge on the real value in the database of the input parameter K, user would be able to try clustering with several values of K, as a simple and rough approach. Unfortunately the fastness of the convergence could not have much effect to overcome the deficiency of capturing in local minimums. The common method is to run the algorithm several times and return the best (not necessary the optimal) clustering found.

## 4 Standard *K*-means clustering

*K*-means algorithm is defined based on the Euclidean distance. The Euclidean distance in a plan is the "ordinary" distance between two points that one would measure with ruler. In N dimension space, the Euclidean distance between two points $P$ and $Q$ is

$$\|P - Q\| = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \quad (1)$$

where $p_i$ (or $q_i$) is the coordinate of $p$ (or $q$) in dimension $i$.

*K*-means uses the concept of Euclidean distance to measure the similarity (or dissimilarity) between each point in the database and the center of the clusters to determine to which cluster the point is better dependent. It tries to separate the data points into an adequate number of clusters in a way that the sum of the Euclidean distances of all points of database to the centriod of their own cluster become minimized (at least locally minimized as mentioned before). This procedure consists of the following steps [13]:

Step 1: Choose $K$ initial cluster centers $z_1(1), z_2(1), \ldots, z_k(1)$ arbitrarily.
Step 2: At the rth iterative step distributes the samples {X} among the $K$ cluster domains, using the relation,

$$x \in S_j(r) \quad if \quad \|x - z_j(r)\| < \|x - z_i(r)\| \quad (2)$$

For all $i$ = 1,2,..., $K$, $i \neq j$, where $S_j(r)$ denotes the set of samples whose cluster center is $z_j(r)$.
Step 3: From the results of Step 2, compute the new cluster centers $z_j(r+1)$, $j = 1, 2, \ldots, K$, such that the sum of the squared distances from all points in $S_j(r)$ to the new cluster center is minimized. Those cluster centers are considered simply the sample mean of $S_j(k)$.

$$z_j(k+1) = \frac{1}{N} \sum_{X \in S_j(k)} X \quad j = 1, 2, \ldots, K \quad (3)$$

where $N_j$ is the number of samples in $S_j(r)$.
Step 4 : if $z_j(r+1) = z_j(r)$, For $j$ = 1,2,...,$K$, the algorithm has converged and the procedure is terminated. Otherwise go to Step 2.

The behavior of the *K*-means algorithm is influenced by the number of cluster centers specified, the choice of initial cluster centers, the order in which the samples are taken, and, of course, the geometrical properties of the data.

## 5 Standard shuffled frog-leaping algorithm

Evolutionary Algorithms (EAs) are a class of search and optimization techniques that have been inspired by the nature

systems, Darwinian Evolution and have been applied widely to solve the complex problems. But it is now well established that pure EAs are not well suited to fine tuning search in complex combinatorial spaces, and also hybrid algorithms using the other techniques can greatly improve the efficiency of search [15,16]. Memetic Algorithms (MA)s are the combination of EAs with the local search methods [17]. MAs are population-based approaches for heuristic search in optimization problems that have been inspired by models of adaptation of a population with individual learning within the lifetime of their members. Its nomination is based on Richard Dawkin's concept of a meme [18], which represents a unit of cultural evolution that can exhibit local refinement.

SFL, a new member in the family of memetic algorithms, is originated from the research of food hunting behaviors of frog. It is based on evolution of memes carried by the interactive individuals and a global exchange of information among themselves [9]. In fact, it combines the benefits of the local search tool of the particle swarm optimization and the idea of mixing information from parallel local searches to move toward a global solution [19]. The shuffled frog-leaping algorithm (SFLA) has been tested on several combinatorial problems and has been found to be efficient in reaching global solutions [20].

The most prominent merit of SFL is its fast convergence speed [10].The SFL algorithm starts with an initial population of $P$ frogs randomly within the feasible space $\Omega$. For $S$-dimensional problems ($S$ variables), a frog $i$ is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})$. Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each containing n frogs (i.e. $P = m \times n$). In this process, the first frog goes to the first memeplex, frog $m$ goes to the *mth* memeplex and frog $m + 1$ goes back to the first memeplex, etc.

Within each memeplex, the frogs with the best and worst fitness values are identified as $X_b$ and $X_w$, respectively. Also, the frog with the global best fitness is identified as $X_g$. Then, a process similar to PSO is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. Accordingly, the position of the frog with the worst fitness is adjusted as follows:

$$\text{Change in frog position}(D_i) = \text{rand}() \times (X_b - X_w) \tag{4}$$

$$\text{New position} X_w = \text{current position } X_w + D_i,\ D_{\max} \geq D_i \geq -D_{\max} \tag{5}$$

where rand() is a random number between 0 and 1; and $D_{\max}$ is the maximum allowed change in a frog's position. If this process produces a better solution, it replaces the worst frog. Otherwise, the calculations in Eqs. (4) and (5) are repeated but with respect to the global best frog (i.e. $X_g$ replaces $X_b$):

$$\text{Change in frog position } (D_i) = \text{rand}() \times \left(X_g - X_w\right) \tag{6}$$

If no improvement becomes possible in this case, then a new solution is randomly generated to replace that frog. The calculations then continue for a specific number of iterations [18]. The pseudo code offer the SFL procedure can be shown as follows [11]:

```
Begin;
  Generate random population of P solutions (frogs);
  For each individual i...p: CALCULATE FITNESS(I);
  Sort the population P in descending order of their fitness;
  Divide P into m memeplexes;
    For each memeplex:
    Determine the best and the worst frogs;
    Improve the worst frog position using Eqs. (4),(5) or (6),(5);   } local search
    Repeat for a specific number of iterations;
    End;
  Combine the evolved memeplexes;
  Sort the population P in descending order of their fitness;
  Check if termination is true;
End;
```

The flowchart of the local search is illustrated in Fig. 2. Accordingly, the main parameters of SFLA are number of frogs $P$; number of memeplexes; number of shuffling iterations; and maximum step size.

## 6 Application of SFLA for clustering VCAs

### 6.1 Clustering validity indices

Cluster validity indices corresponding to the statistical–mathematical functions are used to evaluate the results of a clustering algorithm on a quantitative basis. Ideally, a validity index should take care of the two aspects of partitioning. (1) *Cohesion*: The patterns in one cluster should be as similar to each other as possible. The fitness value of the patterns in a cluster is an indication of the cluster's cohesion or compactness. It is the average of the distance (Euclidean distance) of the cluster members to the centers of clusters. As it is defined as target of clustering algorithms in this study, the smaller the value of fitness function means the better clustering performance. (2) *Separation*: Clusters should be well separated. The distance among the cluster centers (their Euclidean distance) gives an indication of cluster separation. For crisp clustering, some of the well-known indexes available in the literature are the Davies-Bouldin (DB) index, the Dunn's index (DI) and the Calinski–Harabasz index [21,22]. DB index is a function of the ratio of the sum of *within-cluster scatter* to *between-cluster separation*. The scatter within the ith cluster $S_i$, is computed as average of Euclidean distances of all members to the center. The distance between cluster $C_i$ and $C_j$, is defined as$d_{ij} = \left\| z_i - z_j \right\|$, where $z_i$ and $z_j$

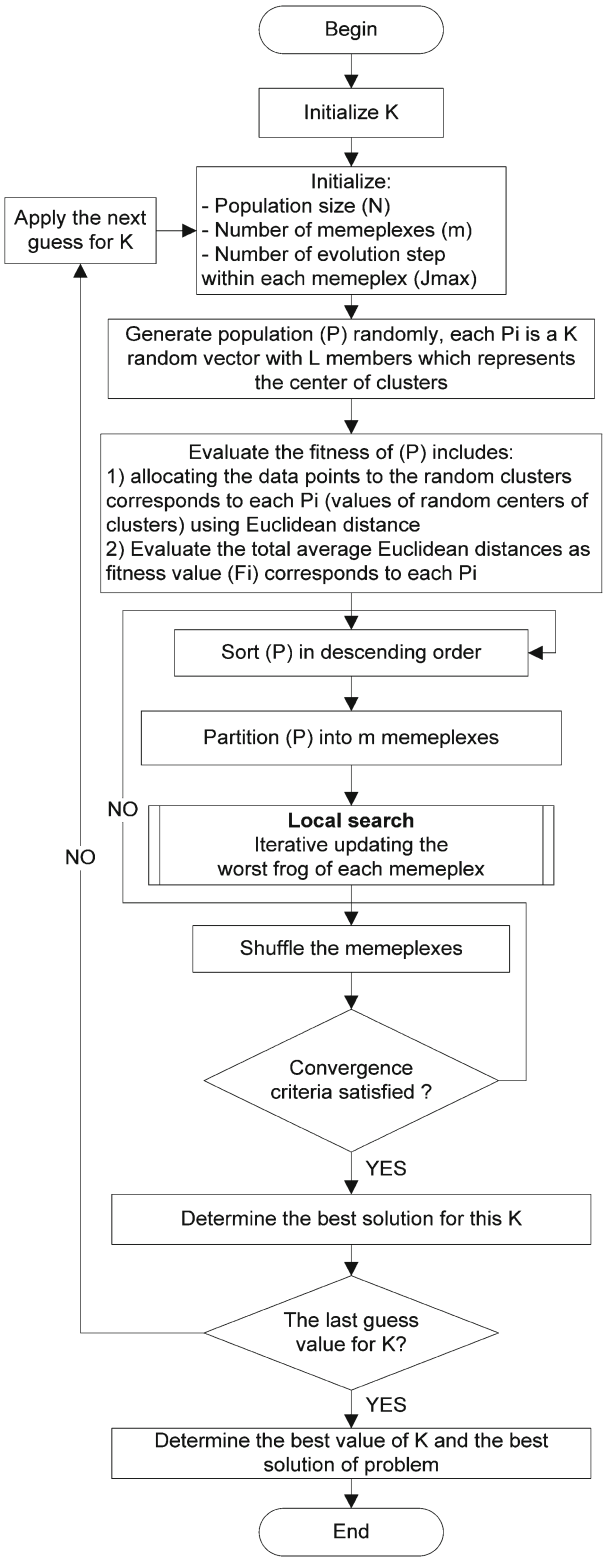


**Fig. 2** Flowchart of Local Search in SFLA

represent the cluster centers. DB index is defined as

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_{i,qt} \tag{7}$$

where $R_{i,qt} = \max_{j,j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\}$. Smaller value for DB index means better clustering performance. In this paper the fitness value and the DB index are used for evaluation the results of clustering.

### 6.2 Selection of the value for K

Both groups of most popular clustering methods, i.e., partitional and hierarchical methods suffer from their intensively dependence to the initial values. Using the memetic algorithms like SFLA helps to rectify this defect by having significant numbers of initial points that can provide a wide area searching space with the ability of multi-directional search.

However, the other defect of partitional algorithms have steel existed in SFL-based data clustering algorithm as follows: (a) The number of clusters can not be determined automatically by the algorithm, and (b) there is no deterministic method or formulation to find $K$, the number of clusters.

On the other hand, it can be easily shown that for a given data set with $N$ members, we could achieve the minimum value for fitness function in $K = N$. In this case the centers of the clusters are exactly the data points themselves. In this manner it is expected that the fitness value decreases monotonically with respect to increasing the $K$. However, in practice finding the correct and exact values for centers of clusters with the existing data clustering methods is impossible as $k$ became too large. In this case, the sum of the created errors from using inexact and improper centers for clusters, makes the total Euclidean distances (or the total fitness value) become large. Hence in practice, the performance of data clustering would decrease with increasing the $K$ more than a defined value. In fact there is a trade off between the accuracy in theory and in practice. On the other hand, using too small values for $K$ would lead to inaccurate clustering. It leads to improper combinations of clusters and unsuitable centroids and enhancement of total Euclidean distances. Having a brief study of the database helps to estimate a good interval for acceptable values for $K$. Choosing several guesses for $K$, an iterative method could help to achieve an optimal value for $K$ in this interval.

### 6.3 The total algorithm

The search capability of SFLA is used here for the purpose of appropriately determining a fixed number of $K$ cluster centers in $R^N$. For suitably clustering the set of N unlabeled vector points, the metric that has been adopted is the sum of the Euclidean distance of the points from their respective cluster centers. As it was mentioned in the previous sections, clustering the VCAs corresponding to the various stress directions and levels and various contingencies which dispose to create

a voltage instability condition in power system is the target of this article. The main steps to accomplish this task are as follows:

1. *Generate database* To increase the chance of having an effective data clustering for VCAs in an actual power system, it is necessary to generate as more as possible the load stress situations and sever contingencies leading to emergency and alert cases of voltage instability and to determine their corresponding VCAs. Simply stated, the accuracy of the final clusters depends directly on how wide the database of the response points could cover the various situations, contingency cases and stress directions. To generate such an adequate database, the following steps must be taken:
   1.1. Generating pre-contingency operating points corresponding to various levels and directions of system stress. For this purpose, $N$ main levels of uniform load stresses are generated by uniformly increasing the basic active and reactive load levels and active generation levels using multiples greater than one (1.1, 1.3, 1.5, 1.7,…). Then at the base point and each of these new main levels, several random direction and value of extra loading are generated using a random vector generation function. For this purpose, a random normal distribution function of MATLAB with mean value near zero and standard deviation factors in the range of 0.01–0.1 have been used.
   1.2. At each of the generated loading levels, several kinds of more probable line outages or generator trips are applied. For this purpose, all of the first-order contingencies including the outage of all lines and also outage of all generators and synchronous compensators have been considered in various load stresses. In addition, many of important second-order contingencies like outage of two significant lines, outage of one important line and one generator have been considered as studied cases.
   1.3. Using a continuation power flow program, maximum loadability of system at each of the situations is measured and the point of instability is determined. It is noticeable that the condition of considering the q-limit of generators has become active when running the CPF program; therefore, both Saddle Node Bifurcation (SNB) and Limit-Induced Bifurcation (LIB) constraints have been taken into account in distinguishing the unstable point [23].
   1.4. At this voltage stability limit point, eigenvalue analysis and calculation of participation factors are accomplished using the approach which has been described in Sect. 3. At the end of this stage, the database is produced. In this database the main components of system characteristics like voltage magnitude and phase at each bus, active and reactive load power, magnitude of active power generation, etc., are the input values, and the participation values of each of the load buses or generators are the output data. These output data include m vectors with n elements: m is the number of generated situations and n is the number of buses of under studying power system.
   1.5. To prepare data for clustering stage, it is required to normalize the data according to the largest element of database. As the bus participation factors are naturally normalized, the normalization step is not required.
2. *Write the Fitness Function appropriate to the clustering purpose and the SFLA,* The fitness function must get $K$ and the random centers of clusters; allocate each data point to a proper cluster based on its minimum Euclidean distance to the center points of clusters (Eq. 2), and afterward return the mean value of the total Euclidean distance of all of the data points from the center of their own clusters as the total fitness value of the current iteration:

$$\text{Fitness} = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n_j}\left(\sum_{i=1}^{n_j}(z_j - x_i)\right)^{1/2}\right) \tag{8}$$

   where $n_j$ is number of members of each jth cluster with center $Z_j$, and m is the total numbers of clusters.
   To generate new worst positions in each memeplex, use the local search algorithm of SFLA illustrated in the flowchart of Fig. 2.
3. *Guess several values for K, the number of clusters based on a glance of the produced data* A glanced identification of database leads to a primary guessed interval for K. The best value for $K$ can be selected by testing different values for $K$, in the interval, comparing the results and choosing the best solution.
4. *Determine N random set for the cluster center* In this problem, each data point is a vector of m normalized participation factors of m buses of power system, so the cluster centers must be vectors of the same dimension ($m$).
5. *Use the SFLA based clustering algorithm* It is illustrated in flowchart of Fig. 3. Evaluation of the clustering performance is done by using DB validity index fitness value

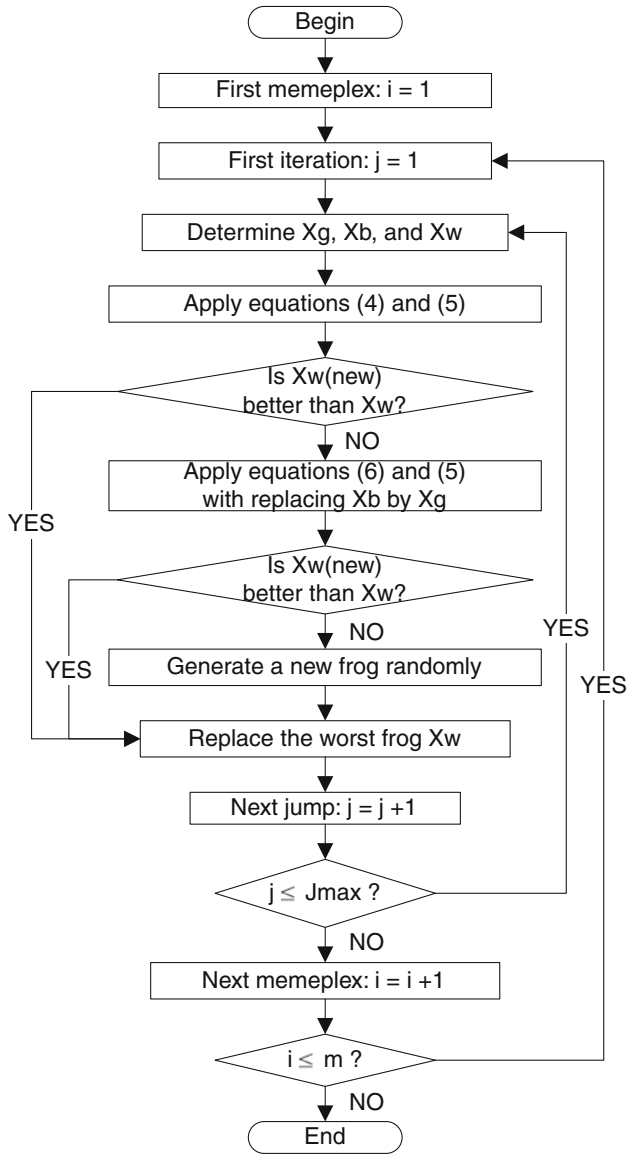


**Fig. 3** Flowchart of Clustering Algorithm using SFLA

at the end of iteration. The algorithm would be stopped when the number of iterations reached to a specific value.

### 6.4 Main parameters of SFLA

The performance of SFLA depends on the values of its main parameters: frog population size, number of memplexes and number of evolution steps. However, the nomination of those parameters is quite experimental and depends on the nature of studied optimization problem. In the other words, there is not any rule to define the optimal values for those parameters.

In [11] the authors have tested the SFL algorithms on two known continuous optimization functions and also on a large-scale discrete problem. In the mentioned paper, different settings were experimented to determine suitable values for parameters to solve the test problems using the SFL algorithm, and finally the recommended experimental values for parameters of SFLA are as follows:

A population of 200 frogs, 20 memeplexes, and ten iterations per memeplex.

The above values have been used in the present article as the initial values for parameters. However, increasing the number of frogs up to 400 led to better results in all of the examined runs. After that, enhancing the population had no significant effect. On the other hand, changing the number of memeplexes and iterations per mepeplex did not lead to better results. Hence, those values have been accepted as the suitable values for SFLA in the present clustering problem.

## 7 Numerical results and analysis

In order to evaluate the performance of SFLA in data clustering, it is necessary to investigate by using several different data sets. To compare the results of SFLA-based clustering with the analytic algorithms, *K*-means, as the most popular analytic algorithm is selected. In addition, it is necessary to compare the performance of proposed algorithm with the other previous evolutionary algorithms. In this study, Genetic algorithm (GA), as one of the most popular and most effective evolutionary algorithms has been considered for this comparison. The parameters of SFLA for this study were defined in Sect. 6 part *D*. The main parameters of GA are selected as follows:

Crossover probability ($CP$) and the mutation probability ($MP$) are set to 0.8 and 0.08, respectively, and the population size is set at 500 offsprings. Number of iterations for SFLA and GA are set to 100.

Furthermore, several sets of artificial data have been generated and applied to prepare the test. The results were satisfactory both from performance and stability (robustness) points of view. In the following the results of applying the aforementioned algorithms on one artificial data set, in addition of one real test data set of participation factors of power system are presented:

a. *Artificial data set* by several running of the normal distribution function with proper narrow standard deviations around mean values which are far enough from each others, well-separated groups of data are generated. Getting 45 runs of normal distribution function around the mean value of zero with standard deviation 0.08, 45 different vectors of 14 data points around this mean value or center point were generated. By repeating the procedure around 2, 4 and 7 as mean values with the same standard deviations, the other groups of

45 vectors of data points were produced. This artificial data set was saved and used to test the algorithm.

Using the three referred algorithms and according to the numerical results, the following points are respectable:

1. In all cases, the numbers of clusters are considered known (4 clusters). Due to using normalized form of data, the initial values for starting the algorithms are considered random values in rang of [0 1]. Table 2 illustrates the results of four consequent runs of *K*-means, SFLA and GA with different initial values which have been selected randomly from the defined accepted range. As can be seen, in *K*-means-based clustering, the results vary significantly even with almost no significant changes in initial values. This is the reason of bad clustering in Cases 2, 3 and 4 of *K*-means algorithm. In contrast, the results of Case 1 are very accurate because of starting the algorithm with a few different initial set. However, the various results taken from SFL-based data clustering show its satisfactory low dependency on initial values, at least if those changes are due to the ordinary selection of initial values from an acceptable and defined range of data. As can be seen in Table 2, the results from four different runs of SFLA lead to correct clustering although the cluster centers are not as accurate as Case 1 of *K*-means algorithm. The robustness and low dependency of SFLA to initial values is indebted to the existence of a swarm of searching particles, frogs, in this algorithm as opposed to having only one particle to search the space in *K*-means algorithm. This provides the ability of multi-directional search in the space. For this reason, improper starting points of some of the frogs could not have any significant effects on achievement of the global optima since the other individuals compensate this defect very well. This is also true for the GA-based data clustering, illustrated in Table 2.
2. Figures 4, 5, 6, 7 show consequently, comparisons between the values of fitness function, DB index, error in determining the centers of clusters (with respect to the actual centers of correct clusters) and number of bad clustered data for SFLA, GA and *K*-means produced by 15 different runs of each algorithm with different initial values. The illustrated parameters are sorted in descending order.
   These figures verify all the aforementioned subjects. It can be observed from Table 2 and Figs. 4 and 5, that the values of DB index and fitness function are much better for SFLA and GA in almost all of the cases in comparison with *K*-means algorithm, although these values in the best cases of *K*-means are slightly better than SFLA and GA. It is true also for the value of mean absolute error of clustering. It is important that the variations of all of the aforementioned indices of data clustering are not too significant in the various runs of GA and SFLA. The results are not as exact as some cases of *K*-means but are acceptable. However, although there are some perfect results among the total runs of *K*-means algorithm, it is not any certification to reach to a proper solution after a defined number of runs, because its intensive dependency on initial values and tendency to cap to the local minimums.
   Also, 100% correct clustering of SFLA and GA, against only 28% of correct answers for *K*-means, confirms the above discussion and shows the much better reliability of the evolutionary algorithms.
3. The values of the aforementioned main indices for GA are very close to the corresponding values for SFLA, but the performance of SFLA seems to be a little better, in almost all of the cases. Furthermore, comparison of CPU times shows that SFL is a faster algorithm than Genetic, illustrated in Table 2.
4. It is an important point that the robustness of SFLA to the initial values is not complete; perhaps main variations in initial values for all or a significant numbers of frogs in one predefined direction would lead to a bad clustering. It is due to making the search space limited and decreasing the dispersion of individual initial values for starting the search. In contrast, if the changes in initial values help to increase the dispersion of initial values and expansion of searching space, it would improve the results. Table 3 contains examples to illustrate this subject. The values are the average over 20 different runs in each case. Taking the Case 2 of SFLA as the base case, in Table 3 for Case 5, the initial positions of all frogs was multiplied by $0.1 \times (i)$, where $i$ is the number of $i$th frog. There were no main changes in the solution, because of increase in dispersion of individual initial values. In Case 6, the initial values of center of clusters 1 and 2 are multiplied by 0.1 for only one half of frogs. It can be seen that as this operation expands the initial space of search, the results are even improved. However, in Case 7, by multiplying the initial values of center of clusters 1 and 3 into 0.05 for all frogs, which leads to a directed change in initial searching space, the resulted clustering became incorrect
5. On the other hand, the CPU time for *K*-means implementation is much more less than for SFLA and GA. It is due to repeating the clustering procedure for a large number of articles in each iteration for SFLA and GA and also having a large number of iterations for confidence of reaching to the global optimization point. However, it is not an important factor in the present study because in VCA determination process, clustering the output data is an off-line stage of the final AIS training and preparation, as mentioned earlier.

**Table 2** Results of clustering the artificial database

| Algorithm | Case | Final fitness value | DB index | Mean values of cluster center | Number of members in Cluster | CPU time (s) |
|---|---|---|---|---|---|---|
| *K*-means | 1 | 0.0029 | 2.0389 | 0.0024 | 45 | 0.9432 |
| | | | | 2.0048 | 45 | |
| | | | | 4.0002 | 45 | |
| | | | | 7.0041 | 45 | |
| *K*-means | 2 | 0.2051 | 1.7934 | 0.0024 | 45 | 0.947 |
| | | | | 2.0048 | 45 | |
| | | | | 2.2609 | 0 | |
| | | | | 5.5021 | 90 | |
| *K*-means | 3 | 0.0143 | 1.5005 | 1.0036 | 90 | 0.925 |
| | | | | 4.0002 | 45 | |
| | | | | 5.3189 | 0 | |
| | | | | 7.0041 | 45 | |
| *K*-means | 4 | 0.0159 | 3.5331 | 0.0024 | 45 | 1.19 |
| | | | | 1.9978 | 14 | |
| | | | | 2.0048 | 31 | |
| | | | | 5.5021 | 90 | |
| SFLA | 1 | 0.0064 | 0.1990 | 0.0039 | 45 | 117.04 |
| | | | | 1.7070 | 45 | |
| | | | | 3.9763 | 45 | |
| | | | | 7.2151 | 45 | |
| SFLA | 2 | 0.0104 | 0.2560 | 0.0320 | 45 | 118.67 |
| | | | | 1.4691 | 45 | |
| | | | | 4.3122 | 45 | |
| | | | | 6.9052 | 45 | |
| SFLA | 3 | 0.0077 | 0.1144 | 0.0377 | 45 | 113.82 |
| | | | | 1.8979 | 45 | |
| | | | | 4.2229 | 45 | |
| | | | | 6.9763 | 45 | |
| SFLA | 4 | 0.0111 | 0.2575 | 0.0244 | 45 | 110.06 |
| | | | | 1.7276 | 45 | |
| | | | | 4.1437 | 45 | |
| | | | | 7.0038 | 45 | |
| GA | 1 | 0.0086 | 0.1057 | 0.0484 | 45 | 152.43 |
| | | | | 1.9748 | 45 | |
| | | | | 3.9381 | 45 | |
| | | | | 7.0645 | 45 | |
| GA | 2 | 0.0088 | 0.1227 | 0.0440 | 45 | 155.18 |
| | | | | 1.7750 | 45 | |
| | | | | 3.9020 | 45 | |
| | | | | 6.8090 | 45 | |
| GA | 3 | 0.0058 | 0.4417 | 0.0512 | 45 | 153.89 |
| | | | | 1.9722 | 45 | |
| | | | | 4.1070 | 45 | |
| | | | | 7.1157 | 45 | |
| GA | 4 | 0.0041 | 0.6120 | 0.0240 | 45 | 154.90 |
| | | | | 2.1023 | 45 | |
| | | | | 4.3416 | 45 | |
| | | | | 6.6920 | 45 | |

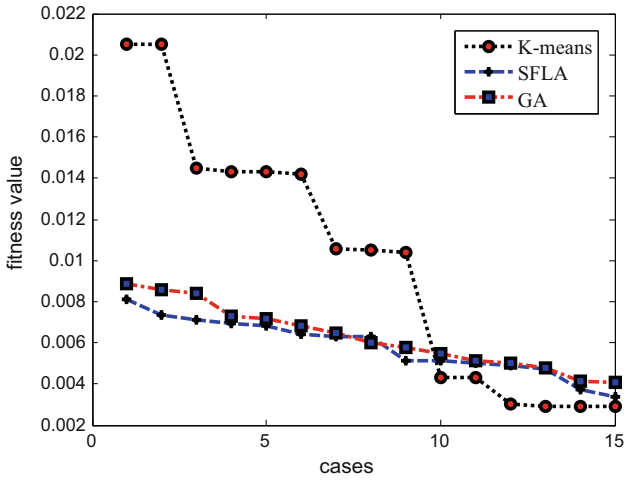


**Fig. 4** Fitness values for different starting points

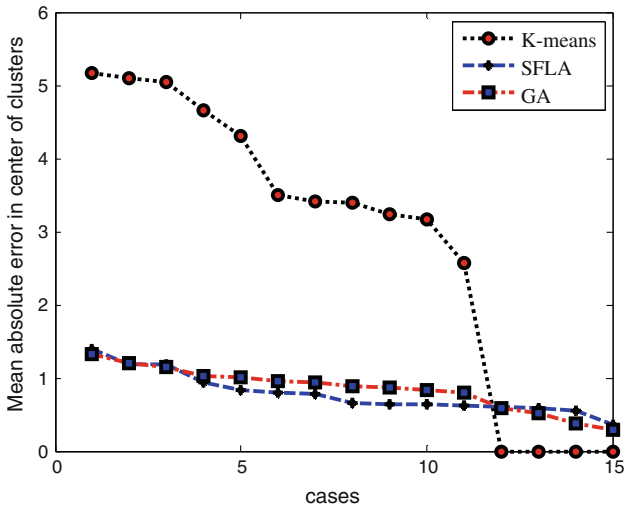


**Fig. 6** Error in center of clusters for different starting points

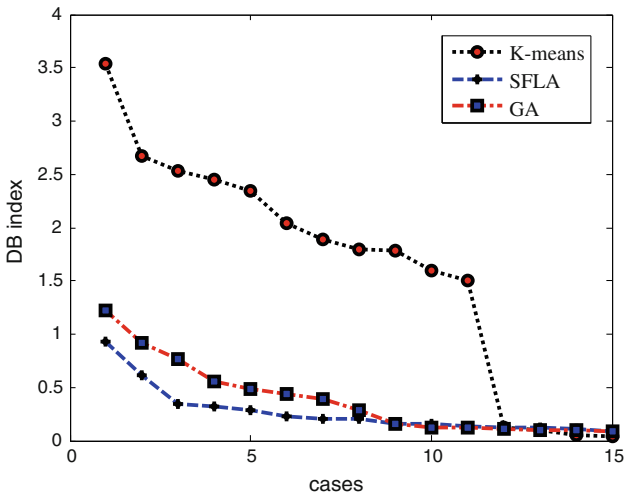


**Fig. 5** DB index for different starting points

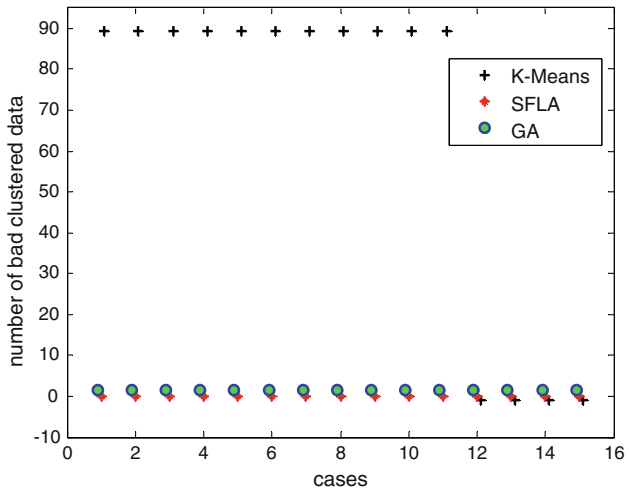


**Fig. 7** Bad clustered data for different starting points

b. *Real data set* To investigate the performance of the proposed algorithm in clustering the PFs and hence the VCAs, a database was built by using 14-bus IEEE test system [24]. This database contains 150 vectors of bus participation factors resulting from applying different cases of stress patterns and contingencies on this test system as was explained in the previous sections. Each participation vector includes 14 elements corresponding to the 14 buses of system. In order to eliminate the influence of randomicity, different clustering algorithms are independently applied 20 times with random initial values and the average results are illustrated in Tables 4 and 5. Contrary to the artificial data case, in the VCA clustering, number of clusters is not known. According to a brief study of the participation factors of all cases, we examined the values, 2, 3, 4 and 5 for the number of clusters. The results of SFL-based data clustering for participation factors are illustrated in Table 4. As can be seen the average values of fitness function and DB Index are optimum in $K = 3$. Therefore, further calculations are based on this value for $K$.

The averages values of fitness function and DB index, the CPU time and the number of members in each cluster over all 20 different runs for SFLA, GA and $K$-means are illustrated in Table 5 for $K = 3$. It is clear that the value of fitness function is slightly better for $K$-means, but the value of DB index implies the advantage of SFLA and GA against $K$-means algorithm in selecting well-separated clusters. It is also noticeable that the results of GA are slightly weaker than SFLA. Due to the importance of having correct and well separated clusters for VCAs and by leaning on the robustness

**Table 3** Results of various changes in initial values for artificial database

| Case | Final fitness value | DB index | Mean values of cluster center | Number of members in cluster |
|---|---|---|---|---|
| 5 | 0.0104 | 0.2610 | 0.1300 | 45 |
| | | | 1.3991 | 45 |
| | | | 4.3729 | 45 |
| | | | 6.9560 | 45 |
| 6 | 0.0092 | 0.2389 | 0.2599 | 45 |
| | | | 2.0104 | 45 |
| | | | 3.9507 | 45 |
| | | | 6.7375 | 45 |
| 7 | 0.0169 | 0.3118 | 0.2677 | 45 |
| | | | 0.3146 | 0 |
| | | | 2.9831 | 90 |
| | | | 6.9466 | 45 |

**Table 4** Comparison between clustering performance by using of various values for $K$

| $K$ | Case | Fitness value | DB | No. of cluster members |
|---|---|---|---|---|
| 2 | Average | 0.0022 | 1.0230 | 129, 21 |
| 3 | Average | 0.0013 | 0.0266 | 129, 20, 1 |
| 4 | Average | 0.0057 | 0.9886 | 129, 14, 4, 3 |
| 5 | Average | 0.0072 | 1.2651 | 129, 11, 5, 3, 2 |

and more reliable capability of SFLA in clustering, the results of this clustering method are selected as the best clusters of VCAs in this case study.

It is seen that for IEEE 14-bus system all of the various situations of emergency and alert cases are dividable into three main clusters that as mentioned in the last sections of this article, lead to simplify the train of AIS for situation classification and help rapid analysis and control of power system. The centers of clusters that have been determined by SFLA-based clustering for IEEE 14 bus test system are illustrated in Table 6. The central points of each cluster represent the participated buses in occurrence of all emergency and alert cases that lead to that cluster. It also shows the average participation factors of those buses and hence the amount of their effectiveness in controlling the ongoing instability.

## 8 Discussion

1. The results of numerical analysis show the better robustness, stability and reliability of evolutionary algorithms and in particular, the SFLA for data clustering in comparing with conventional methods.
2. There is better confidence to reach to the global optimum, using the evolutionary algorithms. It has been proved mathematically that the evolutionary algorithms can always attain the global optimum [25]. By studying the numerical results, this claim would be confirmed due to the almost constant cluster members and slight changes in clustering centers, fitness value and DB index in different runs for both EAs, although the algorithm has been started from different random initial values. In addition, the similarity of the responses of GA and SFLA confirms this claim too.
3. The conclusions show that SFLA is at least as well as GA in data clustering problems. In fact the results of SFLA show its slightly better performance and faster operation in this article. However, the main advantage of SFLA over the other evolutionary algorithms like GA and PSO is its better performance in clustering with equality constrains. It can be shown that this algorithm has more tendencies to protect the convexity of working space during clustering procedure that enables reaching the solutions that could satisfy the linear constraints of the problem, without any manipulations.

## 9 Conclusions

In a large power system, with extreme numbers of stress directions and magnitudes, and probability of occurrence of various kind of contingencies, clustering the VCAs corresponding to different conditions is an important task. It would help to simplify training process of AIS and increase the accuracy of voltage stability assessment and control. In this paper a SFLA for clustering the VCAs have been applied. The results show that SFLA is a suitable and relatively simple method to cluster databases. Investigation of the SFL algorithm on artificial data sets illustrates its good performance and effectiveness. Comparison of this methodology with the most popular analytic algorithm for clustering, $K$-means,

**Table 5** Results of VCA clustering for IEEE 14 bus test system

| Algorithm | Final value of fitness function | DB index | Number of clusters | Number of members of clusters | | | CPU time (s) |
|---|---|---|---|---|---|---|---|
| SFL | 0.0098 | 0.1061 | 3 | 129 | 20 | 1 | 113.624 |
| GA | 0.0105 | 0.1358 | 3 | 129 | 20 | 1 | 154.211 |
| $K$-means | 0.0077 | 1.2712 | 3 | 127 | 15 | 21 | 1.3171 |

**Table 6** Center of clusters generated by SFLA for bus participation factors of IEEE 14-bus test system

| Bus no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cluster center 1 | 0 | 0 | 0 | 0 | 0 | 0.0904 | 0.0753 | 0.0615 | 0.1035 | 0.1102 | 0.1040 | 0.1069 | 0.1134 | 0.1336 |
| Cluster center 2 | 0 | 0 | 0 | 0 | 0 | 0.0898 | 0.0737 | 0.1034 | 0.0621 | 0.1120 | 0.1061 | 0.1100 | 0 | 0 |
| Cluster center 3 | 0 | 0 | 0 | 0 | 0 | 0.0876 | 0.0756 | 0.0693 | 0 | 0.2034 | 0 | 0 | 0 | 0 |

shows its good stability and robustness against changes in initial values, in contrast to strong dependency on initial values of *K*-means algorithm. Also, comparison with GA-based data clustering shows that this algorithm is at least as reliable as GA, but is a faster EA than GA. The satisfactory values of fitness function and especially the DB index show the adequacy of SFLA for clustering purposes.

## References


1. Anderson PM, Le Reverend BK (1996) Industry experience with special protection schemes. In: IEEE/CIGRE committee report; IEEE Trans-On Power Syst., vol II, no. 3, August
2. Loud L, Rousseaux P, Lefebvre D, Van Cutsem T (2001) A time-scale decomposition-based simulation tool for voltage stability analysis. In: Power Tech Conference, Porto, Portugal, IEEE, September
3. Gu X, Canizares CA (2007) Fast prediction of loadability margins using neural networks to approximate security boundaries of power systems. IET Gener Transm Distrib 1(3):466–475
4. Assis TML, Nunes AR, Falcao DM (2007) Mid and long-term voltage stability assessment using neural networks and quasi-steady-state simulation. In: Large engineering systems conference on power engineering, pp 213–217, October
5. Gao B, Morison GK, Kundur P (1992) Voltage Stability Evaluation Using Modal Analysis. IEEE Trans Power Syst 7(4):1529–1542
6. Kundur P (1994) Power system stability and control. EPRI Power System Engineering Series. McGrow-Hill, Inc, New York
7. Morison K, Wang X, Moshref A, Edris A (2008) Identification of voltage control areas and reactive power reserves; an advancement in on-line voltage security assessment. power and energy society general meeting-conversion and delivery of electrical Energy in th 21st century, 2008, IEEE
8. Identification of critical voltage control areas and determination of required reactive power reserves. EPRI final report 1013995, October 2007
9. Eusuff M, Lansey K, Pasha F (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 38(2): 129–154. doi:10.1080/03052150500384759
10. Eusuff M, Lansey K (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resource Plan Manage 129(3):210–225
11. Elbeitagi E, Hegazy T, Griesson D (2005) Comparison among five evolutionary-base optimization algorithms. Adv Eng Inform 19:43–53
12. Zhang X, Hu X, Cu G, Wang Y, Niu Y (2008) An improved shuffled frog leaping algorithm with cognitive behavior. In: Proceeding of the 7th world congress on intelligent control and automation. Chongqing, China, June 2008
13. Tou JT, Gonzalez RC (1974) Pattern recognition principles. Addision-Wesley Publication Company, Reading, MA
14. Chen Zh, Shixiong X (2009) *K*-means algorithm with improved initial center. In: IEEE Second international workshop on knowledge discovery and data mining
15. Culberson J (1998) On the futility of blind search: an algorithmic view of no free lunch. Evol Comput J 6(2):109–128
16. Goldberg D, Voessner (1999) Optimizing global-local search hybrids. In: Proceeding of the genetic and evolutionary computation conference (GECCO-99). Morgan Kaufmann Publishers, San Francisco, pp 220–228
17. Moscato P (1989) On evolution, search, optimization, gas and martial arts: toward memetic algorithms, California Inst. Technol., Pasadena, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826
18. Dawkins R (1976) The selfish GENE. Oxford Univ. Press, New York
19. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceeding of the IEEE international conference on neural network, IEEE Service Center, Piscataway, pp 1942–1948
20. Duan Q, Gupta V, Sorooshian S (1993) Shuffled complex evolution approach for effective and efficient global minimization. J Optim Theory Appl 76:502–521
21. Maulik U, Bandyopadhyay S (2002) Performance evaluation of some clustering algorithms and validity indices. IEEE TRANS. Pattern Anal Mach Intell 24:12
22. Halkidi M, Vazirgiannis M (2001) Clustering validity assessment: finding the optimal partitioning of a data set. In: Proceeding of IEEE international conference on data mining, pp 187–194
23. Avalos RJ, Canizares CA, Millano F (2008) Equivalency of continuation and optimization methods to determine saddle-node and limit-induced bifurcations in power systems. IEEE Trans. Circuits Syst
24. University of Washington, College of Engineering, Electrical Engineering, Power Systems Test Case Archive. http://www.ee.washington.edu/research/pstca. Accessed 18 Jan 2010
25. Ombach J (2008) Stability of evolutionary algorithms. J Math Appl 342:326–333