An Efficient Shuffled Frog Leaping Algorithm for Clustering Of Gene Expression Data

Athul Jose
PG scholar
Department of CSE
Bannari Amman Institute of Technology

M Pandi
Asst Professor (Sr Gr)
Department of CSE
Bannari Amman Institute of Technology

ABSTRACT

Shuffled frog-leaping algorithm is an evolutionary algorithm, which uses a stochastic search method that mimics natural biological evolution and the social deeds of species.These evolutionary algorithms are developed to find a new optimum solution for optimization problems that cannot be solved by gradient based mathematical methods. The generation of frog leaping algorithm is drawn from two other search techniques: the local search of the particle swarm optimizationand the competitiveness mixing of information of the shuffled complex evolution technique. This paper then introduces a new parameter for acceleration of searching into the formulation of the original shuffled frog leaping algorithm to create a modified form of the algorithm for effective clustering of gene expression data.

Keywords

Memeplexes,Evolutionary algorithms,shuffling, memetic, evolution, swarm.

1. INTRODUCTION

The optimization of systems and processes is very important to the efficiency and economics of science and engineering domains. Rigorous or approximate mathematical search methods are used to solve optimization problems. Careful approaches were usedin linear programming, dynamic programming or branch-and-bound techniques, andinteger programming to arrive at the optimum solution for moderate-size problems. Optimizing real-life problems of the scale often encountered in engineering practice is much more challenging because of the huge and complex solution space. Finding accurate solutions for these problems turn out to be NP-hard. As the number of decision variables increases, these complex problems require an exponential amount of computing power and time.To overcome these problems researchers have proposed approximate evolutionary-based algorithms [11] as a means to search for near-optimum solutions.

Clustering is the process of grouping similar objects such that the same group consists of most similar objects [12] [5]. The main approaches for clustering methods are partitional and hierarchicalmethods. Grouping of gene expression data will help in finding the expression levels of thousands of genes at a time.

The first evolutionary based technique [3]introduced in the literature was the genetic algorithm by Holland in 1975. In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid local optima, various genetic algorithm improvements have been proposed during the past 10 years, with the latest and perhaps most promising technique being the shuffled frog-leaping (SFL) algorithm (Eusuff and Lansey 2003, Elbeltagi et al. 2005) [1] [10]. This paper first presents the SFLA method and then introduces arandom point$X_1$to the original formulation to proposean Extended Shuffled Frog-Leaping (ESFL) algorithm. To determine the best limit of values for this new parameter, a parametric study was conducted. Various example problems are considered to examine the effectiveness of the acceleration constraint on solution excellence and convergence speed of the ESFL algorithm.

In this paper, the fitness function used is variance. The fitness of an organism is how much it can reproduce before it dies. Let X= {*x*1, *x*2 ..., *xn*} be a set of elements, in which each element is a vector of *d*-dimension. Each gene is an element $x \in X$, and *xi* is the value of its expression level under experimental condition *i*. Given a subset *Y* = {*y*1, *y*2... *ym*} of *X*, let *c*(*Y*) denote the centroid of *Y* and let its variance be

$$VAR(Y) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{d}\left(y_{i,j} - c(y)_j\right)^2 \quad (1.1)$$

2. SHUFFLED FROG-LEAPING ALGORITHM

Pseudo code for Shuffled Frog Leaping Algorithm

```
Begin;
Generate random population of P solutions (individuals);
For each individuali Є P: calculate fitness (i);
Sort the whole population P in descending order of their fitness;
The population P is divided into m memeplexes;
For each memeplex;
Determine the best and worst individuals;
Using Eqs. 2.1 and 2.3 improve the worst individual position;
Repeat for a given number of iterations;
End;
Evolved memeplexes are combined;
Sort the population P in descending order of their fitness;
Check if termination=true;
End;
```

The shuffled frog-leaping algorithm is a memetic metaheuristicalgorithm that is designed to seek a global optimal solution by performing a heuristic search. This algorithm is based on the evolution of memes carried by individuals and a global exchange of information among the population (Eusuff and Lansey). It combines the advantage of local search tool of the particle swarm optimization (Kennedy and Eberhart), and the idea of mixing information from parallel local searches to move toward a global solution.

The SFL algorithm has been undergone many modifications [9][10] and has been tested on several combinatorial problems and found to be efficient in finding global solutions. (Eusuff and Lansey 2003) [2]. The SFL algorithm involves a population of possible solutions defined by a set of frogs (i.e. solutions) that is partitioned into subsets which may be referred as memeplexes. Each memeplexes are considered as the cultures of frogs, and each performing a local search. In each memeplex, the individual frogs hold ideas that may be subjectively change with the ideas of other frogs, which will lead to evolution through a process of memetic evolution. After a number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process (Liong and Atiquzzaman). The local search and the shuffling processes continue until convergence criteria are satisfied. X.H. Luo et al, [9] in 2009 proposed a modified shuffled frog-leaping algorithm to solve travelling salesman problem.

The SFLA is derived from a virtual population of frogs in which individual frogs are equivalent to the GA chromosomes, and represent a set of solutions. Each frog is distributed to a different subset of the whole population called a memeplex. An independent local search is conducted for each frog memplex, which is called memeplex evolution. After completing a defined number of memetic evolutionary steps, frogs are shuffled among memeplexes enabling frogs to interchange messages among different memeplexes and ensure that they move to a most favourable position, like particles in PSO [8]. Shuffling and Local search continues until defined convergence criteria are met. SFLA have demonstrated effectiveness in a number of global optimization problems difficult to solve using other methods, like water distribution and groundwater model calibration problems [2].

First, 'P' number of frogs is created randomly as an initial population. For S-dimensional problems, each frog i is represented by S variables as Xi= (xi1, xi 2, ...,xiS). The frogs are sorted in descending order based on fitness value; and the frogs are then sorted in a descending order. The entire population is then divided into m memeplexes, each containing n number of frogs (i.e. P=m×n). In this procedure, the first memeplex will get the first frog, second memeplex will get the second frog, frog m goes to the mth memeplex, andfrog m+1 goes to the first memeplex again, and so on. Within each memeplex, the frogs with the best fitness is identified as Xband the frog with worst fitness is identified as and Xw. The frog with the global best fitness is considered as Xg. To improve the frog with worst position, an evolution process is applied in each cycle. The frog with worst position is adjusted as follows:

Change in frog position,

$(D_i) = \text{rand}().(X_b\text{-}X_w)$ (2.1)

New position of frog,

X_w= current position Xw+ Di; (2.2)

$(D_{max} >= D_i >= -D_{max})$

Where rand( ) is a random number between 0 and 1; and Dmax is the maximum allowed change in a frog's position. If this process produces a better frog (solution), it replaces the worst frog. Otherwise, the calculations in equations (2.1) and (2.2) are repeated with respect to the global best frog (i.e. Xg replaces Xb).

Change in frog position,

$(Di) = rand() . (Xg\text{-} Xw)$ (2.3)

If no improvement becomes possible in this latter case, then a new solution is randomly generated to replace the worst frog with another frog having any arbitrary fitness.The calculation then continues up toa given number of evolutionary iterations within each memeplex (Eusuff and Lansey 2003) [2].

3. AN EFFICIENT SHUFFLED FROG-LEAPING ALGORITHM (ESFLA)

In the SFL algorithm, each memeplex is allowed to evolve independently to locally search at different regions of the solution space. Additionally, shuffling all the memeplexes and re-dividing them again into a new set of memeplexes results in a global search through changing the information between memeplexes. The SFL algorithm attempts to balance between a wide search of the solution space and a deep search of promising locations that are close to a local optimum. As expressed by equation (2.1), each individual frog (solution) in a memeplex is trying to change its position towards the best frog within the memeplex or the overall best frog. As shown in this equation, when the difference in position between the worst frog Xw (i.e. the frog under evolution) and the best frogs (Xb or Xg) becomes minute, then the change in frog Xw's position will be very small, and thus it might stagnate at a local optimum and cause premature convergence. To overcome such problems, this paperproposes that the right-hand side of equation (2.1) be added with X_1 which is apointgenerated randomly.

3.1 Initial population

For an*S*-dimensional problem,an initial population of *P* frogs is created randomly. A frog *i* is represented by S variables, such as $Fi = (f_{i1}, f_{i2} \ldots, f_{is})$

3.2 Sorting and distribution

Based on the fitness values frogs are sorted in descending order, then entire population is divided into *m* memeplexes, each containing *n*numberfrogs (i.e., $P = m \times n$). , the first memeplex will get the first frog, second memeplex will get the second frog, the *m* frog to the *m*th memeplex, andthe *m*+ 1th frog will be assigned again to the first memeplex and the process continues.

3.3 Memeplex evolution

Within each memeplex, the frogs with the best fitness is identified as Xb and the frog with worst fitness is identified as and Xw. The frog with the global best fitness is considered as Xg. To improve the worst solution, an equation similar to PSO is used to update the worst solution, e.g., Eqs. (2.1) and (2.2):

Change in frog position,

$Di=rand() .(Xb–Xw)+ X_1$ (3.1)

New position of frog,

Xw= current position *Xw*+ *Di* (3.2)

$(D\text{max} >= Di >= -D\text{max})$

Where*rand*() is a random number between 0 and 1;$X_1$is a random point generatedand *D*max is the maximum displacement of frog.Assigning a large value by using randomly generated value X_1at the beginning of the evolution process will accelerate the global search by allowing for a bigger change in the frog's position and accordingly will widen the global search area. Then, as the evolution process continues and a promising location is identified, the search will focus the process on a deeper local search as it will allow the frogs to change its positions.It replaces the worst frog, if this process produces a better solution. If Eqs. (3.1) and (3.2) do not improve the worst solution, and then make use of Eq. (3.3):

Change in frog position

$(Di)= rand() .(Xg- Xw)$ (3.3)

If this also doesn’t improve the frog’s position, then a new solution is randomly generated.

3.4 Shuffling

After a defined number of memeplex evolutions, allfrogs from each memeplexes are collected, and sorted in descending order based on their fitness. Step 3.2 divides frogs into different memeplexes again, and then step 3.3 is done.

3.5 Terminal condition

If a global solution is obtained or a fixed iteration number is reached, the algorithm stops.

4. EXPERIMENTAL RESULTS

The experimental results comparing the ESFLA clustering algorithm with other typical stochastic algorithms, such as SFLA algorithm, ACO algorithm [6],and the simulated annealing approach [7] are provided with 4 real life dataset solutions. The experiments are carried out on a Pentium IV system with, 512-MB RAM and have coded with Matlab R2012 software.

4.1 Vowel Data

The data consist of 871 Indian Telugu vowel sound data. These were expressedby three male speakers in the age group of 30–35 years as a consonant–vowel–consonant context. The dataset has three main features, F1, F2, and F3, which correspondsto the vowel formant frequencies of first, second and third respectively and six overlapping classes {δ, a, i, u, e, o}. The number of clusters is therefore chosen to be 6 for these data [13].

4.2 Iris Data

This is the Iris dataset, is abest known database to be found in the pattern recognition. The dataset contains three classes of 50 instances each.In which each class denotes to a kind of iris plant. Here one class will be linearly separable from other two; the latter are not linearly separable from each other. There are a total of 150 instances with four numeric attributes in iris dataset and there is no missing attribute value. The attributes in iris dataset are; petal length, sepal length, petal width, and sepal widthall calculated in centimetre.

4.3 Crude Oil Data

These overlapping data has 56 data points, three classes and five features. So, the number of clusters is chosen to be 3 for this dataset [13].

4.4 Wine Data

This is the wine dataset, which is also taken from MCI laboratory. These chemical analyses of wines grown in the same region in Italy but derived from three different cultivators are the result data. The examination determined the quantities of 13 constituents found in each of the three types. There are 13 numeric attributes with 178 instances in wine dataset.

4.5 Thyroid Diseases Data

This dataset categories N=215 samples of patients suffering from three human varieties of thyroid diseases, namely euthyroid, hyperthyroidism, and hypothyroidism. 150 individuals are tested euthyroid thyroid; 30 patients were having hyperthyroidism thyroid, and 35 patients suffered hypothyroidism thyroid. All individuals are characterized by the result of five laboratory tests, like total serum thyroxin, total serum tri-iodothyronine, serum tri-iodothyronine resin uptake, serum thyroid-stimulating hormone (TSH), and increased TSH after injection of TSH-releasing hormone.

The comparison of results for each dataset is based on the best solution found in ten distinct runs of each algorithm and the convergence processing time taken to attain the best solution. The solution quality is also given in terms of the average and worst values of the clustering metric (F_{avg}, F_{worst}, respectively) after ten different runs for each of the four algorithms. F is the performance of clustering method.

For Vowel data (Table 1), the ESFL clustering algorithm attains the best value of 148022.482417 in 90% of runs.SFL algorithm attains the best value 148815.726432 in 90% of runs ACO,and SA [4] provided the best values in 80% of runs. In addition, the SFL clustering algorithm performed better than other algorithms in terms of theprocessing time required (67.72) while ESFLA took 69.30.

For clustering problem on Iris dataset, results given in Table 2show that the SFLA provide the optimum value of 96.550842 while ESFLA provide 96.104923. The SFLA, ACO and ESFLA were able to find the optimum nine times as compared to that of five times obtained by SA. The SFLA required the least processing time (32.11) and ACO (33.72) and ESFLA (34.18) were close to optimum.

For Crude Oil dataset, the ESFL clustering algorithm attains the best value of250.214367 in 90% of total runs and SFLA attains the best value of 251.534997 in 90% of total runs and ACO, and SA attain the best value of 253.564637 and 253.763548 in 80% of total runs respectively. The processing time required by SFL is less than the other algorithms (14.37) while ESFLA (15.96) took a little more.

The result obtained for the clustering problem Wine dataset is given in Table 4. The ESFLA found the optimum solution of 16212.808466and SFLA got optimum solution as 1,6279.539104, and the ACO and SA[7] methods provide 1,6530.533807. The ESFLA, SFLA, ACO, and SA methods found the optimum solution in all their ten runs. The execution time taken by the SFL algorithm is less than the other algorithms.

The ESFL algorithm for the Human Thyroid Disease dataset provides the optimum solution of 10103.825503 and SFLA provides optimum solution of 10,107.71535 to thisproblem, with a success rate of 90% during ten runs. In terms of the processing time, the SFLA performed better than other clustering algorithms. (Table5).

Table I. Result obtained by the four algorithms for ten different runs on vowel data

Method	**Function Value**			**CPU time (S)**
	F_{best}	**$F_{average}$**	**F_{worst}**	
ESFLA	148022.482417	148022.765883	148023.452735	69.30
SFLA	148815.726432	148815.937224	148817.834347	67.72
ACO	148837.736634	148837.768828	148837.937878	73.65
SA	149357.634587	149436.175420	149749.549362	79.46

Table II. Result obtained by the four algorithms for ten different runs on iris data

Method	**Function Value**			**CPU time (S)**
	F_{best}	**$F_{average}$**	**F_{worst}**	
ESFLA	96.104923	96.106285	96.112903	34.18
SFLA	96.550842	96.5525835	96.568257	32.11
ACO	97.100777	97.171546	97.808466	33.72
SA	97.100777	97.134625	97.263845	95.92

Table III. Result obtained by the four algorithms for ten different runs on crude oil data

Method	**Function Value**			**CPU time (S)**
	F_{best}	**$F_{average}$**	**F_{worst}**	
ESFLA	250.214367	250.357490	252.103298	15.96
SFLA	251.534997	251.684314	253.028164	14.37
ACO	253.564637	2,541.808972	256.645938	14.98
SA	253.763548	2,546.532078	258.211847	24.74

Table IV. Result obtained by the four algorithms for ten different runs on wine data

Method	**Function Value**			**CPU time (S)**
	F_{best}	**$F_{average}$**	**F_{worst}**	
ESFLA	16212.808466	16212.953411	16212.953952	55.62
SFLA	16279.539104	16279.539104	16279.539104	53.18
ACO	16530.533807	16530.533807	16530.533807	68.29
SA	16530.533807	16530.533807	16530.533807	57.28

Table V. Result obtained by the four algorithms for ten different runs on thyroid data

Method	**Function Value**			**CPU time (S)**
	F_{best}	**$F_{average}$**	**F_{worst}**	
ESFLA	10103.825503	10104.103772	10107.632187	98.41
SFLA	10107.71535	10108.09731	10111.53499	95.73
ACO	10111.827759	10112.126903	10114.819200	102.15
SA	10111.827759	10114.045265	10115.934358	108.22

Flowchart of Efficient Shuffled Frog Leaping Algorithm

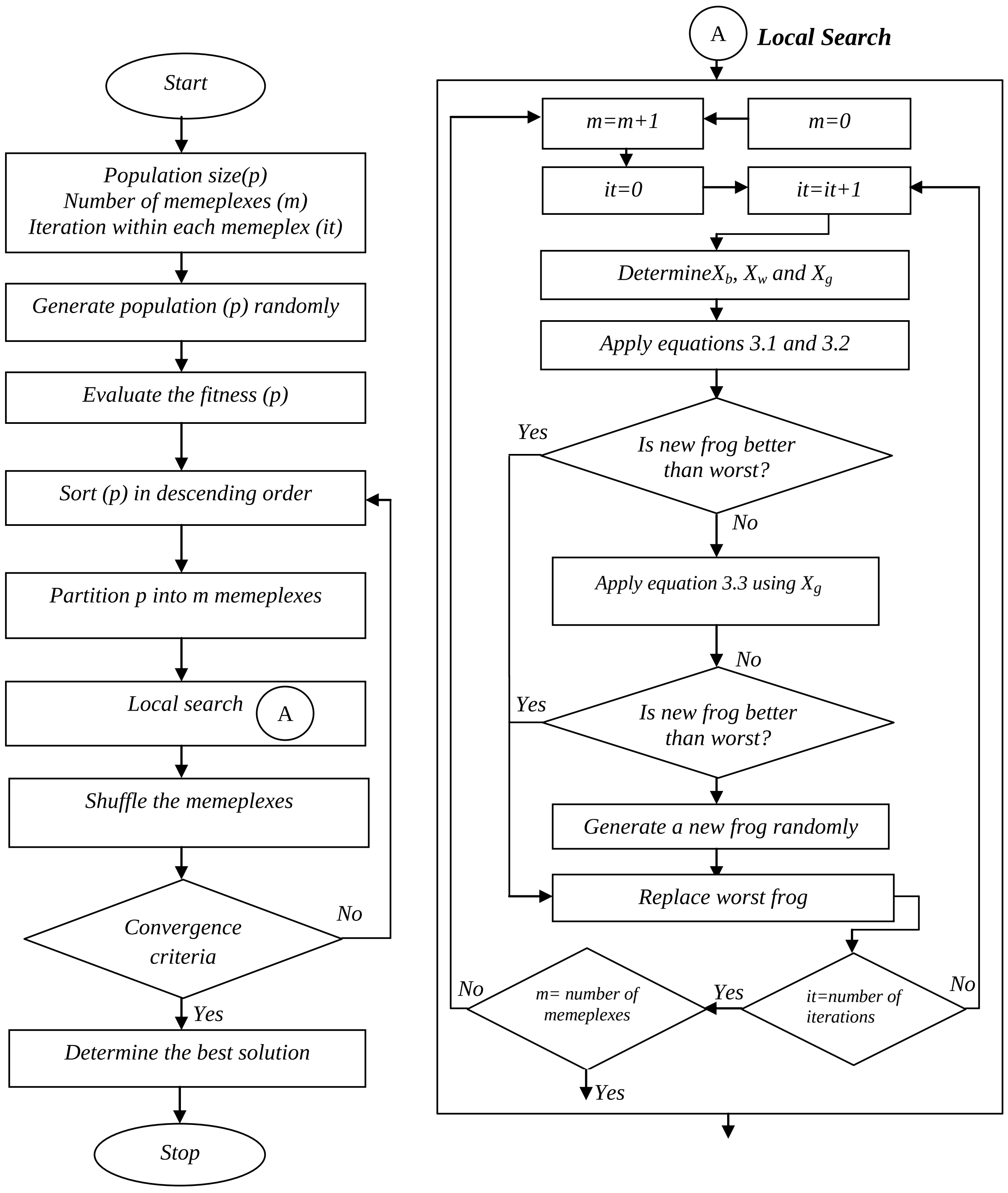
Start
Population size(p)
Number of memeplexes (m)
Iteration within each memeplex (it)
Generate population (p) randomly
Evaluate the fitness (p)
Sort (p) in descending order
Partition p into m memeplexes
Local search
A
Shuffle the memeplexes
Convergence criteria
No
Yes
Determine the best solution
Stop
A
Local Search
m=m+1
m=0
it=0
it=it+1
Determine X_b, X_w and X_g
Apply equations 3.1 and 3.2
Yes
Is new frog better than worst?
No
Apply equation 3.3 using X_g
No
Yes
Is new frog better than worst?
Generate a new frog randomly
Replace worst frog
No
m= number of memeplexes
Yes
it=number of iterations
No
Yes

Fig 1: Efficient Shuffled Frog Leaping Algorithm

5. CONCLUSION

The ESFL algorithm is an optimization algorithm used for solving the clustering problem. Swarm based approach for optimization is followed in SFLA.This algorithm is based on evolution of memes carried out by global exchange of information by interactive individuals. The Shuffled Frog Leaping algorithm can be applied for data clustering when the number of clusters is known apriori. In the proposed ESFL algorithm, to accelerate the global search by a big change in the frog's position,a random point$X_1$is used.

The ESFL algorithm is then compared with SFL algorithm and other stochastic algorithms like Simulated Annealing and Ant Colony Optimization. The algorithm is tested and simulated using various data sets like vowel dataset, iris dataset, crude oil dataset, wine dataset and thyroid dataset. Among these algorithms ESFL algorithm is providing better results and in case of computation time ESFLA is taking slightly higher amount of time than Shuffled Frog Leaping Algorithm.

As a future work, it is possible to reduce the execution time and increase the global search area for getting more optimal solutions.

As a future work new modifications can be performed on equations to improve the global search area. This can be done by changing the frog's position accordingly. The ranking and evolution process can be modified to get a better solution.

6. REFERENCES

[1] Eusuff M, Lansey K, Pasha F. 2006. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. EngOptim 38(2):129–154.

[2] Eusuff MM, Lansey K.E. 2003. Optimizing of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manage 129(3):210–225.

[3] Elbeltagi E, Hegazy T, Grierson D 2005 Comparison among five evolutionary-based optimization algorithms. J Adv Eng Inform 19:43–53.

[4] Shokri Z, Selim, Al-Sultan K. 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognition 24 (10):1003–1008.

[5] Sung CS, Jin HW. 2000.A Tabu-search-based heuristic for clustering. Pattern Recognition 33:849–858.

[6] Shelokar PS, Jayaraman VK, Kulkarni BD .2004. An ant colony approach for clustering. Anal ChimActa 509:187–195.

[7] Gungor Z, Unler A .2007.K-harmonic means data clustering with simulated annealing heuristic. Appl Math Comput 184:199–209

[8] Lovbjerg, M., 2002. Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality. MSc thesis, Aarhus University, Denmark.

[9] X.H. Luo, Y. Yang, X. Li, 2009.Modified shuffled frog-leaping algorithm to solve traveling salesman problem, Journal of Communications 30 (7) 130–135.

[10] J.P. Luo, M.R. Chen, X. Li, .ICIEA2009. A novel hybrid algorithm for global optimization based on EO and SFLA, in: Proceedings of the Fourth IEEE conference on Industry electronics and applications, Xi'an, China, 2009, pp. 1935–1939.

[11] Emad E, Tarek H, Donald G. 2005. Comparison among five evolutionary-based optimization algorithms. Adv Eng Inform 19:43–53.

[12] Spath H, 1989. Clustering analysis algorithms. Ellis Horwood,Chichester, UK

[13] Kuo RI, Wang HS, Hu T-L, Chou SH. 2005. Application of ant K-means on clustering analysis. Comput Math Appl 50:1709–1724.