Performance analysis of firefly algorithm for data clustering

Hema Banati and Monika Bajaj*

Department of Computer Science,
University of Delhi,
Delhi 110007, India
E-mail: banatihema@hotmail.com
E-mail: mbajaj48@gamil.com
*Corresponding author

Abstract: Extraction of relevant information from web is gaining immense significance. This depends upon the efficacy of methods employed to represent, organise and access the information. The most important technique that is used to organise the data is clustering. Clustering problem refers to partitioning unlabeled data objects into certain number of clusters with the objective of maximum homogeneity within cluster and heterogeneity between the clusters. The paper studies viability of firefly algorithm for clustering. It incorporates the flashing behaviour of fireflies to achieve the optimal solution. The performance of the proposed algorithm FClust is compared with particle swarm optimisation (PSO) and differential evolution (DE) algorithms with respect to varied statistical criteria using artificial and benchmark datasets. A detailed convergence behaviour of algorithms is studied using run length distribution. The experimental results prove that the proposed algorithm performs better in terms of speed and success rate as compared to PSO and DE.

Keywords: firefly algorithm; data clustering.


Biographical notes: Hema Banati completed her PhD (2006) after her Masters in Computer Applications (MCA) both from Department of Computer Science, University of Delhi, India. Currently, she is an Associate Professor in the Department of Computer Science, Dyal Singh College, University of Delhi. She has over 18 years of teaching experience to both undergraduate and postgraduate classes. Over the past decade, she has been pursuing research in the areas of web engineering, software engineering, human computer interaction, multiagent systems, e-commerce and e-learning. She has many national and international publications to her credit.

Monika Bajaj is a Research Scholar in Computer Science Department at University of Delhi. Her research interests include web engineering, human computer interfaces and e-commerce. She has many national and international publications to her credit.

Copyright © 2013 Inderscience Enterprises Ltd.
1 Introduction

The explosive growth of information with the popularisation of the web makes it difficult to extract relevant information. The relevance of extracted information depends upon the efficacy of methods employed to represent, organise and access the information. One such method employed is clustering that organises the data in turn facilitating the extraction of relevant information. Clustering problem refers to partitioning the unlabeled data objects into certain number of clusters. Since the objective of clustering is to achieve first homogeneity within cluster, i.e., objects belonging to the same cluster should be as similar as possible and second heterogeneity between the clusters, i.e., objects belonging to different clusters should be as different as possible, the algorithmic task can be stated as an optimisation problem (Brucker, 1978). The objective functions of clustering problems are usually non-linear and non-convex so some clustering algorithms may fall into local optimum. Moreover, they possess exponential complexity in terms of number of clusters and become an NP-hard problem when the number of clusters exceeds (Welch, 1983). Clustering is usually performed when no information is available concerning the membership of data items to predefined classes. For this reason clustering is traditionally seen as a part of unsupervised learning.

The simplest and the most popular clustering algorithm is k-means algorithm. It is very efficient, due to its linear time complexity but the deterministic local search used in algorithm, may converge to the nearest local optima. Various meta heuristics and swarm intelligence-based evolutionary algorithms such as tabu search (Al-Sultan, 1995), simulated annealing (Bandyopadhyay et al., 1998; Selim and Al-Sultan, 1991), genetic algorithms (GA) (Bandyopadhyay and Maulik, 2002a, 2002b; Bandyopadhyay et al., 1995, 1999), ant colony optimisation (ACO) (Shelokar et al., 2004), particle swarm optimisation (PSO) (Karthi et al., 2008) and differential evolution (DE) (Paterlini and Krink, 2006) are applied to clustering problem. However these algorithms successfully applied heuristic and evolutionary approaches to avoid convergence to local optima with limited success rate.

Firefly algorithm (FA) is a recently introduced nature inspired approach for solving non-linear optimisation problem (Yang, 2008, 2009). The algorithm is based on the behaviour of social insects (fireflies) where each firefly has own agenda and coordinates with other fireflies in the group (swarm) to achieve the same. Various researchers in the literature study and incorporate the flashing behaviour of fireflies for optimisation purposes. Banati and Bajaj (2011) simulate the attraction system of real fireflies that guides the feature selection procedure. The algorithm combines the rough set theory with FA algorithm to find the optimal reducts. Höning (2010) present an approach that applies the firefly in the context of task graph scheduling. Jati and Suyanto (2011) addresses how to apply FA for travelling salesman problem (TSP). Khadwilard et al. (2011) develop FA for solving job shop scheduling problem and Senthilnath et al. (2011) measure the performance of FA with respect to supervised clustering problem.

The paper presents a new clustering approach that formulates the flashing behaviour of fireflies with the objective function of the clustering problem. The paper recasts the work of Lukasik and Zak (2009) for continuous constrained optimisation tasks to suit the need for data clustering. The study shows the applicability of standard FA to cluster arbitrary data and developed FA-based clustering algorithm FC1ust. The proposed algorithm is centroid evolutionary based, thus we compare the performance of FC1ust with two centroid evolutionary approaches viz. PSO and DE. The performance of each
algorithm is evaluated with respect to two statistical criteria, i.e., TWR and VRC (Paterlini and Krink, 2006) mentioned in Section 2.

The remaining sections of the paper are organised as follows. Section 2 gives an overview of the clustering problem and statistical criteria used for evaluation. Section 3 describes the FClust algorithm for clustering. Section 4 describes the experimental set-up regarding the algorithmic parameters, benchmark problems, and run schedule. Section 5 reports the main results, and finally, Section 6 presents and analysis of these results and concludes the study.

2 The problem statement

Clustering is the process of partitioning a given set of \( n \) points into \( K \) groups or clusters based on some similarity (distance) metric.

Let \( O = \{\tilde{o}_1, \tilde{o}_2, \ldots, \tilde{o}_N\} \) be a set of \( N \) objects to be clustered and each object is represented as \( \tilde{o}_i = \{o_{i1}, o_{i2}, \ldots, o_{id}\} \) where \( o_{id} \) represents a value of object \( i \) at dimension \( d \).

The goal of clustering algorithm is to determine a set of \( K \) partitions \( C = \{C_1, \ldots, C_k\} \)

Then:

\[
C_i \neq \emptyset, \text{ for } i = 1, \ldots, k, \\
C_i \cap C_j = \emptyset, \text{ for } i = 1, \ldots, k, j = 1, \ldots, k \text{ and } i \neq j \\
\text{and } \bigcup_{k=1}^{K} C_k = O.
\]

Each cluster is represented by a cluster centre then \( Z = \{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_k\} \) is a set of cluster centres to which data objects are assigned. In this study we will use Euclidean metric as distance metric,

\[
\tilde{z}_i = \frac{1}{n_i} \sum_{o_i \in C_i} \tilde{o}_i
\]

(1)

where \( n \) is number of objects in cluster.

2.1 Statistical criteria

The following statistical measures are used to evaluate the results generated by different clustering algorithms (Paterlini and Krink, 2006).

2.1.1 Minimisation of trace within criteria (TRW)

This criteria equation (2) is based on pooled within groups scatter matrix \( W \). The pooled within scatter matrix \( W \) is defined as \( \sum_{k=1}^{K} W_k \), where \( W_k \) is the variance matrix of the data object allocated to cluster \( C_k \), where \( k = \{1, \ldots, K\} \).

\[
W_k = \sum_{i \in C_k} \left( \tilde{o}_i^k - \tilde{0}_k^k \right) \left( \tilde{o}_i^k - \tilde{0}_k^k \right)^T
\]
where $\bar{O}_i$ indicates the $i^{th}$ data object in cluster $C_k$ and $n_k$ is the number of objects in cluster $C_k$, and $\bar{O}_k = \left(\sum_{i=1}^{n_k} \bar{O}_i^k\right)/n_k$ is the vector of the centroid for the cluster $C_k$.

2.1.2 Maximisation of variance ratio criteria

This criteria is based on pooled within groups scatter matrix $W$ and between group scatter matrixes $B$. The between scatter matrix $B$ is defined as equation (3),

$$B = \sum_{k=1}^{K} n_k (\bar{O}_k^i - \bar{O}) (\bar{O}_k^i - \bar{O})^T$$

where $\bar{O} = \left(\sum_{i=1}^{N} \bar{O}_i\right)/N$.

$$VRC = \frac{\left(\frac{\text{trace}(B)}{K-1}\right)}{\left(\frac{\text{trace}(W)}{N-K}\right)}.$$  \hspace{1cm} (4)

3 Firefly algorithm

FA (Yang, 2008) is inspired by biochemical and social aspects of real fireflies. Real fireflies produce a short and rhythmic flash that helps them in attracting their mating partners (regardless of their sex) and also serves as protective warning mechanism. FA formulates this flashing behaviour with the objective function of the problem to be optimised.

Let us consider the optimisation problem where the task is to minimise objective function $f(x)$ for $x \in S \subset \mathbb{R}^n$, i.e., find $x^*$ such as

$$f(x^*) = \min_{x \in S} f(x)$$ \hspace{1cm} (5)

The swarm of $n$ fireflies solve the above mentioned problem iteratively and $x_t$ represents a solution for a firefly $i$ at iteration $t$, whereas $f(x_t)$ denotes its fitness. The fitness of each firefly is determined by the landscape of the objective function. Furthermore, this fitness value determines the attractiveness of every $i^{th}$ member in the swarm and represented as a light intensity $I_i$. Initially, all fireflies are dislocated in $S$ (randomly or employing some deterministic strategy). Each firefly finds its matting partner based on attractiveness of other fireflies in the space and move towards that partner in order to improve its fitness. The next position of firefly $i$ at iteration $t + 1$ is determined as per equation (6) that considers two factors first, the attractiveness of other swarm members with higher light intensity, i.e., $I_j > I_i$, $\forall_{j \neq i}$, which is varying across distance and second a fixed random step vector $u$.

$$x_i(t + 1) = (1 - \beta)x_i(t) + \beta x_j(t) + \mu$$  \hspace{1cm} (6)

where $\beta$ is the attractiveness of firefly $j$. It states how strong it attracts firefly $i$ in the swarm and calculate using equation (7).
\[ \beta = \beta_0 e^{-\gamma r} \]  
(7)

where \( r_{ij} = d(x_i, x_j) \), a Euclidean distance between two firefly \( i \) and \( j \). In general, \( \beta_0 \in [0, 1] \), describes the attractiveness at \( r = 0 \), i.e., when two fireflies are found at the same point of search space \( S \). The value of \( \gamma \in [0, 10] \) determines the variation of attractiveness with increasing distance from communicated firefly. It is basically the light absorption coefficient and generally \( \gamma \in [0, 10] \) could be suggested (Yang, 2008, 2009).

The proposed algorithm FClust recasts the above mentioned firefly algorithm scheme to improve the performance of clustering algorithm. Each firefly \( i \) is represented in \( k \) dimensions where each dimension represents the centroid of cluster and moves its position in order to achieve the objective functions stated in equations (2) and (4). The algorithm will allow a firefly to move towards its mating partner only if there is an improvement in its intensity otherwise it will move with randomised steps \( \mu_i \) in the space with probability \( p \). The movement of a firefly \( i \) is attracted to another (brighter) firefly \( j \) is determined by equation (6) with \( \beta_0 \) and \( \gamma = 1 \) as suggested by Yang (2009).

**Figure 1** FClust: clustering algorithm

---

**Input:**
Create randomly an initial population of \( n \) fireflies within \( k \)
dimensional search space \( x_{ik}, i = 1, 2, \ldots, N \) and \( k = 1, 2, \ldots, K \)
Evaluate the fitness of the population \( f(x_i) \) which is directly
proportional to light intensity \( I_i \)
Algorithm’s parameters \( \beta_0 \) and \( \gamma \), \( p \)

**Output:**
Obtained firefly \( i \) with minimum \( f(x_i) \)

**while** (\( t < \text{maxiteration} \))

**for** \( i = 1 \) to \( n \) **do**

**for** \( j = 1 \) to \( n \) **do**

if \( f(x_j) < f(x_i) \) then

find most attractive \( j \) for \( i \) using equation (7)

end if

end for \( j \)

**for** \( k = 1 \) to \( K \) **do**

find \( x_{(t+1),k} \) using equation (6)

end for \( k \)

Evaluate \( x_{(t+1),k} \) using equations (2) and (4)

if \( f(x_{(t+1),k}) < f(x_k) \) then

Update \( x_k \) with \( x_{(t+1),k} \)

else if random \( \sim U(0, 1) \Rightarrow p \)

\[ x_{t+1} = x_t + u_i \]

end if

end for \( i \)

end while
The randomised step $\mu_i$ moves between lower and upper bounds. Yang (2008) proposed to use $\min \mu_i = -0.5 \alpha$ and $\max \mu_i = 0.5 \alpha$ with $\alpha \in [0, 1]$ being algorithm’s parameter. The algorithm FClust given in Figure 1 generates $\mu_i$ according to equation (8) with $\alpha = 1$.

$$\mu_i = \alpha (\text{random} \sim U(0, 1) - 0.5) \quad (8)$$

Figure 1 presents the FClust clustering algorithm. The following sections focus on the experimental set and the performance analysis of the proposed algorithm.

4 Experimental set-up

The experimental study evaluates the performance of proposed algorithm FClust by comparing the results with PSO (Karthi et al., 2008), and differential equation (Paterlini and Krink, 2006). All the algorithms are executed in python 2.7 language and all the experiments are performed on a Pentium IV 400 MHz personal computer. The effectiveness of the algorithms is measured on the basis of following criterion:

1. mean best fitness value of TRW, VRC as mentioned in equations (2) and (4)
2. percentage of number of runs (i.e., success %) that reach best known objective function value over 20 simulations
3. run length distribution (RLD) as proposed by Hoose and Stutzl (2004).

The effectiveness of evolutionary algorithms depends upon the generation of initial population. Therefore all algorithms run for 20 times and allow executing 100,000 fitness evaluations in each run for each dataset. The parameter settings for PSO, DE and FClust are depicted in Table 1 and the six datasets (two artificial and four real world datasets) used for experiments are described below.

Table 1 Parameter settings used for the PSO, DE, and FClust algorithms

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>50</td>
<td>Population size</td>
<td>50</td>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>$W$</td>
<td>0.1</td>
<td>Crossover rate</td>
<td>0.9</td>
<td>$p$</td>
<td>0.98</td>
</tr>
<tr>
<td>$C_1, C_2$</td>
<td>0.01</td>
<td>Factor</td>
<td>0.1</td>
<td>$\mu_{0.5}$</td>
<td>1</td>
</tr>
</tbody>
</table>

4.1 Artificial datasets

We have generated two simulated datasets $s1$ and $s2$ using a random number generator that produced multivariate normal distributions set of objects. The description of datasets is given below:

- **Dataset $s1$:** This dataset is composed of 300 objects with three features distributed over $K =$ six clusters (50 objects in each cluster). The data was generated using mean $\mu_1 = [1, 1, 1], \mu_2 = [5, 5, 5], \mu_3 = [9, 9, 9], \mu_4 = [-3, -3, -3], \mu_5 = [-7, -7, -7], \mu_6 = [-11, -11, -11]$ and sigma $\Sigma_1 = I, \Sigma_2 = I, \Sigma_3 = I, \Sigma_4 = I, \Sigma_5 = I, \Sigma_6 = I.$
Performance analysis of firefly algorithm for data clustering

• Dataset s2: This dataset is composed of 300 objects with three features distributed over \( K = 6 \) clusters (50 objects in each cluster). The data was generated using mean \( \mu_1 = [1, 1, 1], \mu_2 = [5, 5, 5], \mu_3 = [9, 9, 9], \mu_4 = [-3, -3, -3], \mu_5 = [13, 13, 13], \mu_6 = [17, 17, 17] \) and sigma \( \Sigma_1 = I, \Sigma_2 = I, \Sigma_3 = I, \Sigma_4 = I, \Sigma_5 = I, \Sigma_6 = I \).

4.2 Benchmark datasets

In addition to artificial dataset, the following well known benchmark datasets are considered to evaluate the performance of the algorithms.

a Fisher’s iris dataset: The dataset consists of 150 samples of three iris flowers, i.e., \( k = 3 \), viz. Each object is defined by four attributes: sepal length, sepal width, petal length and petal width. This is obtained from the UCI repository of machine learning database (Fisher, 1950).

b Wine dataset: This dataset contains chemical analysis of 178 wines, derived from three different cultivars, \( k = 3 \). Each object is defined by 13: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoids phenols, proanthocyaninism, colour intensity, hue, OD280/OD315 of diluted wines and proline. It is also available in the public domain of UCI repository of machine learning databases (Aeberhard et al., 1992).

c Thyroid dataset: The dataset categories 215 samples of patients suffering from three human thyroid diseases, \( k = 3 \) as: euthyroid, hyperthyroidism, and hypothyroidism patients (Coomans et al., 1978).

d Wisconsin breast cancer dataset: This dataset consists of 683 objects characterised by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses (Street et al., 1993).

5 Performance analysis

5.1 Mean best fitness value: artificial datasets

The mean best fitness values over 20 runs in the analysis with respect to artificial datasets are reported in the Table 2. The results are compared with PSO and DE algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Criteria</th>
<th>Best known values</th>
<th>PSO</th>
<th>DE</th>
<th>FClust</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>TRW</td>
<td>865.71</td>
<td>865.71 (0)</td>
<td>865.71 (0)</td>
<td>865.17 (0)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>2,846.77</td>
<td>2,836.36 (33.44)</td>
<td>2,846.77 (0)</td>
<td>2,846.77 (0)</td>
</tr>
<tr>
<td>s2</td>
<td>TRW</td>
<td>889.76</td>
<td>889.76 (0)</td>
<td>900.99 (31.98)</td>
<td>889.76 (0)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>2,756.11</td>
<td>2,756.11 (0)</td>
<td>2,751.04 (15.59)</td>
<td>2,756.11 (0)</td>
</tr>
</tbody>
</table>
The results of mean fitness in Table 2 show that for the dataset of type $s_1$ all three algorithms (PSO, DE, FClust) converge to the same fitness value for TRW measure whereas for VRC measure, FClust and DE both achieve the optimal value with zero standard error as compare to PSO that attains the mean fitness of 2,836.36 with standard error of 33.44.

The mean best fitness for the dataset of type $s_2$ for TRW and VRC measure reveals superior performance of FClust and PSO. Both algorithms achieve the optimal value with zero standard error as compared to DE that attains the mean fitness of 900.99 and 2,751.04 with standard error of 31.98 and 215.59 for TRW and VRC measure respectively.

5.2 Mean best fitness value: real world datasets

The mean best fitness values over 20 runs in the analysis with respect to benchmarked datasets are reported in the Table 3. The results clearly show that for iris and cancer datasets all three algorithms (PSO, DE, FClust) converge to the same fitness value (i.e., the best known value) for both TRW and VRC measures in every run.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Criteria</th>
<th>Best known values</th>
<th>PSO</th>
<th>DE</th>
<th>FClust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>TRW</td>
<td>7,885.14</td>
<td>7,885.14 (0)</td>
<td>7,885.14 (0)</td>
<td>7,885.14 (0)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>561.63</td>
<td>561.63 (0)</td>
<td>561.63 (0)</td>
<td>561.63 (0)</td>
</tr>
<tr>
<td>Cancer</td>
<td>TRW</td>
<td>19,323</td>
<td>19,323 (0)</td>
<td>19,323 (0)</td>
<td>19,323 (0)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>1,026.26</td>
<td>1,026.26 (0)</td>
<td>1,026.26 (0)</td>
<td>1,026.26 (0)</td>
</tr>
<tr>
<td>Wine</td>
<td>TRW</td>
<td>2,370,689.68</td>
<td>2,370,689.68 (0)</td>
<td>2,370,689.68 (0)</td>
<td>2,370,689.68 (0)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>561.81</td>
<td>561.56 (0.6767)</td>
<td>561.81</td>
<td>561.81</td>
</tr>
<tr>
<td>Thyroid</td>
<td>TRW</td>
<td>28,560.15</td>
<td>28,651.65 (96.67)</td>
<td>28,642.22 (90.51)</td>
<td>28,618.33 (88.10)</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>131.83</td>
<td>130.71 (1.13)</td>
<td>131.48 (0.63)</td>
<td>131.50 (0.55)</td>
</tr>
</tbody>
</table>

Regarding the wine dataset all three algorithms achieve the optimal value of TRW measure in every run. Whereas for VRC measure DE and FClust both achieve the target value but PSO has a mean value 561.56 with a standard error of (0.6766) which is acceptable at 95% of confidence level (i.e., $[561.269, 561.862]$ that includes the optimal value 561.63).

The results of thyroid dataset reveal superior performance of FClust as compared to DE and PSO. However, all the three algorithms are not able to reach the optimal value in every run for both TRW and VRC measure, but FClust maintains the mean fitness of TRW value with 28,618.33 among all algorithms with a lower standard error of 88.10. For VRC measure all the algorithms are close to each other but FClust performs well by achieving mean fitness value of 131.50 with standard error (0.55) as compared to DE with mean value 131.48 and SE (0.63) and PSO with 130.71 mean value and standard error of 1.13.
5.3 Success percentage

The success rate in terms of percentage of number of runs (i.e., success %) that reach best known objective function value over 20 simulations is reported in Table 4 for artificial datasets and for benchmarked datasets in Table 5.

Table 4 Percentage of number of runs (i.e., success %) for artificial datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Criteria</th>
<th>PSO</th>
<th>DE</th>
<th>FClust</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>TRW</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>s2</td>
<td>TRW</td>
<td>100</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>100</td>
<td>90</td>
<td>100</td>
</tr>
</tbody>
</table>

5.3.1 Success rate for artificial dataset

The results shown in Table 4 for s1 dataset indicate that both FClust and DE reaches the optimal value of TRW and VRC in every run whereas PSO gained 100% success rate for TRW but for VRC it is able to achieve 90% success rate out of 20 individual runs.

Regarding dataset s2, the FClust and PSO achieve the 100% success for TRW and VRC measure in every run whereas DE gained 85% success for TRW and 90% success for VRC criterion.

5.3.2 Benchmark datasets

The results shown in Table 5 indicate that PSO, DE and FClust perform well for iris and cancer datasets. Regarding wine dataset both FClust and DE reach the optimal value of TRW and VRC in every run whereas PSO gained 100% success for TRW but for VRC it is able to achieve 16 times the target value out of 20 individual runs. However, the results of thyroid dataset depict that all the three algorithms are fail to achieve 100% success for both TRW and VRC measure but FClust achieve the higher success rate by attaining 11 times the target value of TRW as compare to DE which is able to attain only nien times and PSO that achieve only six times from 20 simulations. For VRC measure FClust attain 11 times optimal value whereas DE and PSO attain ten and five times respectively.

Table 5 Percentage of number of runs (i.e., success %) for benchmark datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Criteria</th>
<th>Best known value</th>
<th>PSO</th>
<th>DE</th>
<th>FClust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>TRW</td>
<td>7,885.14</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>561.63</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Cancer</td>
<td>TRW</td>
<td>19,323</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>1,026.26</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Wine</td>
<td>TRW</td>
<td>2,370,689.68</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>561.81</td>
<td>80</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Thyroid</td>
<td>TRW</td>
<td>28,560.15</td>
<td>30</td>
<td>45</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>VRC</td>
<td>131.83</td>
<td>25</td>
<td>50</td>
<td>55</td>
</tr>
</tbody>
</table>
5.3 Run length distribution

An important measure to evaluate the quality of the solutions provided by the stochastic
algorithms is the number of function evaluations. Thus, we use the methodology RLD
proposed by Hoos and Stutzle (2004) to analyse the convergence behaviour of algorithms
with respect to the objective under consideration. RLD’s are plotted for all the datasets.
RLD plot shows the convergence of the algorithms with respect to the number of function
evaluations and also indicates the probability of reaching a pre-specified objective
function value over specified number of functional evaluations. The probability value
(success rate) is the ratio between the number of runs finding a solution of certain quality
and the total number of runs. RLD plots for the artificial datasets are shown in the
Figure 2 to Figure 5 and for benchmark datasets are shown in the Figure 6 to Figure 13.

5.3.1 Artificial datasets

RLD for the three stochastic approaches on artificial datasets s1 for TRW metric is shown
in Figure 2. The distribution shows that FClust has the fastest convergence behaviour
with respect to the best known value whereas DE and PSO has a slow increasing rate
towards 1.0 probability of finding the best known value. The FClust algorithm is able to
achieve the best known value with the probability of 1.0 within 200 evaluations as
compared to DE and PSO that achieve within 500 evaluations. Figure 3 shows the RLD
for VRC measure of s1 dataset. The results show that FClust achieve the best known
value with probability of 1.0 within first 100 evaluations. However, DE also attained the
0.1 probability but take 500 evaluations whereas PSO attain the 0.9 probability to solve
the problem.

RLD for datasets s2 for TRW metrics is shown in Figure 4. The distribution shows
that FClust has the fastest convergence behaviour with respect to the best known value
whereas PSO has a slow increasing rate towards 1.0 probability of finding the best known
value and DE has 0.85 probability to find the target value. Figure 5 shows the RLD for
VRC measure of s2 dataset. The results show that FClust find the best known value
with probability of 1.0 within 100 evaluations. However, PSO also attains the 0.1 probability
but with slow increasing rate whereas DE has 0.9 probability of solving the problem.

5.3.2 Benchmark datasets

RLD for the three stochastic approaches on iris dataset for TRW metrics is shown in
Figure 6. The distribution shows that all the algorithms hit the optimal value within
100 function evaluations but with different probability. FClust has 0.95 probability of
achieving the best known value in first 100 function evaluations and the algorithm is able
to find a solution of required quality with a probability of 1.0 in first 200 function
evaluations. However, DE has 0.55 probability of finding the optimal solution in first
100 evaluations and achieves the success with 0.1 probability within 300 evaluations.
PSO achieve the target with 0.75 probability in first 100 evaluations but has slow
increasing rate to reach best known value with probability 0.1. Figure 7 shows the RLD
of iris dataset for VRC measure. All the algorithms reach the best-known value within
100 function evaluations. However, FClust and DE find a solution of required quality
with a probability of 1.0 in first 200 function evaluations. But FClust has 0.85 probability of achieving the success within 100 evaluations as compare to DE and PSO that has 0.55 and 0.65 probability respectively.

RLD of cancer dataset for TRW metrics is shown in Figure 8. The distribution shows that all the algorithms are able to hit the optimal value within 100 function evaluations. Whereas FClust has 0.90 probability to find the best known value within 100 function evaluations in comparison of DE and PSO. Figure 9 shows the run length distribution of cancer dataset for VRC measure. The performance of FClust is remarkable as it finds the optimal solution within 100 evaluations in each run. Whereas DE is able to achieve the optimal value with 1.0 probability within 300 evaluations and PSO find the solution with 1.0 probability within 1,000 evaluations.

Figure 10 shows the RLD of wine dataset for TRW criteria. All the algorithms increase slowly to reach the best known value with 1.0 probability. However, FClust and DE both hit the best known value with 1.0 probability within 300 function evaluations, but FClust performs well in first 100 evaluations as compare to DE and PSO. The RLD of wine dataset for VRC measure is shown in Figure 11. The graph shows that both FClust and DE both are able to obtain the solution with 1.0 probability whereas PSO finds the best-known value with a probability of 0.80. FClust has a 0.95 probability of achieving the target value within 200 function evaluations in comparison to DE with 0.85 probability and PSO with 0.75 probability.

RLD for each algorithm on thyroid dataset for TRW metrics are shown in Figure 12. All three algorithms move slowly towards optimal value. The distribution shows that both DE and FClust are able to hit the optimal value within 100 function evaluations with 0.05 probability as compare to PSO which does not find the optimal value. However, with this dataset both DE and PSO find the optimum results with 0.45 and 0.30 probability respectively whereas the FClust performs well and achieves the maximum probability of 0.55 to find best-known value. Figure 13 shows the RLD of thyroid dataset for VRC measure. The distribution shows that this is the only case where DE initially hits the optimal value within 100 function evaluations with 0.05 probability as compare to FClust and PSO. But FClust compensates it by attaining the maximum probability of finding best-known value among all three algorithms. PSO and DE find a solution of required quality with a probability of 0.25 and 0.5 respectively whereas FClust achieves the 0.55 probability of finding a solution of required quality which is remarkable performance.

Interesting observations can be made from the RDL’s for TRW measure. All three algorithms hit the best-known value within 100 evaluations and have increasing success rate to reach optimal value for the dataset. DE and FClust performed well in most of the cases but FClust ensures the maximum probability of finding best-known value among all three algorithms. It is also observe that the convergence of the FClust is fast for most of the benchmark datasets such as iris, cancer and wine. Another interesting observation from the RLDs for VRC metric is that the performance of FClust is remarkable with respect to benchmark datasets. The FClust algorithm has higher probability to find the required optimal value in comparison to DE and PSO. Thus, the convergence of the proposed FClust algorithm for the VRC measure is faster as compare to DE and PSO for most of the benchmark problems.
6 Conclusions

The paper presents clustering algorithm FClust that incorporates the basic behaviour of firefly algorithm to improve the solution of clustering problem. The performance of the proposed algorithm FClust is evaluated by comparing the results with PSO and DE using artificial and benchmark datasets. The experimental study shows that the FClust has higher probability to achieve the optimality as compared to PSO and DE. Run length distribution analysis has been carried out to study the stagnation behaviour and convergence speed of algorithms. The results of the RLD plots indicate that the convergence of the FClust is fast as compared to PSO and DE and the performance of FClust is remarkable.

RLD plots for the artificial datasets are shown in the Figure 2 to Figure 5.

Figure 2  RLD plots for TRW measure of $s1$ dataset (see online version for colours)

Figure 3  RLD plots for VRC measure of $s1$ dataset (see online version for colours)
Figure 4  RLD plots for TRW measure of $s2$ dataset (see online version for colours)

Figure 5  RLD plots for VRC measure of $s2$ dataset (see online version for colours)

RLD plots for benchmark datasets are shown in the Figure 6 to Figure 14.

Figure 6  RLD plots for TRW measure of iris dataset (see online version for colours)
Figure 7  RLD plots for VRC measure of iris dataset (see online version for colours)

![RLD plots for VRC measure of iris dataset](image)

Figure 8  RLD plots for TRW measure of cancer dataset (see online version for colours)

![RLD plots for TRW measure of cancer dataset](image)

Figure 9  RLD plots for VRC measure of cancer dataset (see online version for colours)

![RLD plots for VRC measure of cancer dataset](image)
Figure 10  RLD plots for TRW measure of wine dataset (see online version for colours)

Figure 11  RLD plots for VRC measure of wine dataset (see online version for colours)

Figure 12  RLD plots for TRW measure of thyroid dataset (see online version for colours)
Figure 13 RLD plots for VRC measure of thyroid dataset (see online version for colours)

References


