Computer Science • 24(3) 2023 https://doi.org/10.7494/csci.2023.24.3.5116

I.M.T.P.K. Ilankoon
U.S. Samarasinghe
M.K.A. Ariyaratne
R.M. Silva

FINDING PLAYING STYLES OF BADMINTON PLAYERS USING FIREFLY ALGORITHM-BASED CLUSTERING ALGORITHMS

Abstract *Cluster analysis can be defined as applying clustering algorithms with the goal of finding any hidden patterns or groupings in a data set. Different clustering methods may provide different solutions for the same data set. Traditional clustering algorithms are popular, but handling big data sets is beyond the abilities of such methods. We propose three big data clustering methods based on the firefly algorithm (FA). Three different fitness functions were defined on FA using inter-cluster distance, intra-cluster distance, silhouette value, and the Calinski-Harabasz index. The algorithms find the most appropriate cluster centers for a given data set. The algorithms were tested with nine popular synthetic data sets and one medical data set and are later applied on two badminton data sets with the intention of identifying the different playing styles of players based on their physical characteristics. The results specify that the firefly algorithm could generate better clustering results with high accuracy. The algorithms cluster the players to find the most suitable playing strategy for a given player where expert knowledge is needed in labeling the clusters. Comparisons with a PSO-based clustering algorithm (APSO) and traditional algorithms point out that the proposed firefly variants work in a similar fashion as the APSO method, and they surpass the performance of traditional algorithms.*

Keywords firefly algorithm, clustering, intra- and inter-cluster distance, badminton

Citation Computer Science 24(3) 2023: 427–450

1. Introduction

The fundamental concept of clustering is the grouping of data points in creating partitions based on their similarity. If two things are similar in some ways, they often share other characteristics. Almost everything that we perceive is in the form of clusters. A cluster is a set of similar data points or a set of points that are more similar to each other than two points in other clusters. This is classified as an unsupervised learning technique; the key difference from other machine-learning techniques is that clustering does not have a response class after grouping observations. A human needs to visually look at clusters and optionally associate the meaning to each cluster. The ultimate prediction is the set of clusters themselves. This technique only works with data that is in a numeric form; this means that any categorical variable needs to be converted into a numeric variable. Clustering is essential in many applications, and the segmenting of clusters is important for future reasoning and decision-making.

Clustering is essential to many fields, including medicine, engineering, sports, bioinformatics, image processing and transformation, and many more, and it has emerged as an effective solution to various problems. It is highly used because of some special properties that are inherent to clustering algorithms such as their scalability, high dimensionality, ability to deal with unstructured data, and interpretability.

Many algorithms have been introduced for clustering, such as k-means, fuzzy c-means, k-medoids, x-means, and the Nelder-Mead simplex method. However, clustering big data sets is beyond the abilities of these conventional methods. The highlighted drawbacks of conventional clustering methods include the following: they are highly dependent on initial parameter values, unable to escape from local optimum points, unable to detect numbers of clusters automatically, and sensitive to outliers and noisy data; they also feature relatively low scalability in general. These drawbacks cannot be tackled for bigger data sets; therefore, the need for new algorithms that can offer accurate and efficient clustering while minimizing the existing drawbacks is without doubt.

There are only a few clustering problems that are polynomially bounded. Such algorithms have the property where a method exists where the number of computational steps is bounded by a polynomial in the length of the input of the problem [9]. On the other hand, many clustering problems that we encounter in real life are NP-hard in nature [34] where their computational times grow exponentially with the size of the input.

Nature-inspired meta-heuristics are flexible methods that have the ability to solve complex (NP-hard) and discrete optimization problems. The clustering problem that goes under unsupervised learning in machine learning can be considered to be an optimization problem and can be solved with the help of meta-heuristics. Being stochastic and imitators of undoubtedly well natural optimization tasks, these meta-heuristics can handle NP-hard problems without considering the continuity nor the differentiability of a problem as well as without getting trapped in local optimum solutions. Nature-inspired algorithms are mainly categorized as evolutionary, swarm-

intelligence, and physically inspired algorithms. The algorithms search a solution space to find the optimum solution using a population of individuals with the exploration and exploitation properties that are inherent in them. Each individual in a population is considered to be a potential solution to a problem. In each iteration, the quality of each individual is measured; after a predefined number of iterations, the best solution is taken as the final result. Considering the NP-hard properties that are exhibited in clustering problems, meta-heuristics have been applied in research in order to obtain better clustering solutions. One of the main considerations in this regard is to find proper algorithms as well as proper evaluations of the qualities of the clusters; this is still open for research.

The firefly algorithm (a member of the family of nature-inspired algorithms) has attracted much attention since it was developed, and it has been applied in many applications. This is a population-based optimization algorithm that mimics a firefly's attraction to flashing lights. A firefly is a winged beetle that is commonly known as a lightning bug due to the charming light that it emits. This light is used to attract mates and prey. FA is naturally a multi-modal algorithm; therefore, it is suitable for structural engineering problems – especially when we need to prepare some engineering alternatives in multi-modal problems. In fact, a population of fireflies show characteristic luminary flashing activities that function to attract partners, communicate with each other, and warn against the risk of predators. Being inspired from these activities, Xin-She Yang formulated this method to solve optimization problems [37].

In this research, the firefly algorithm can find better clusters because of several reasons:

1. FA optimization seems more promising than other meta-heuristics (such as particle swarm optimization) in the sense that FA can deal with multi-modal functions more naturally and efficiently.
2. In addition, particle swarm optimization and some other meta-heuristics are just a special class of the firefly algorithm.
3. Since FA is a nonlinear system, it has the ability to automatically subdivide a whole swarm into multiple sub swarms. This is because short-distance attraction is stronger than long-distance attraction, and the division of a swarm is related to the mean range of the attractiveness variations.

Apart from such apparent features, it is easy to implement, is efficient, is adaptive, has low computation cost as compared to other meta-heuristics, and is highly able to solve complex and discrete optimization problems, producing near-optimal solutions. The firefly algorithm considers the natural flashing behaviors of fireflies to attract mates. The canonical algorithm assumes three facts about firefly behavior:

1. attraction of fireflies is gender-independent;
2. attractiveness is proportional to brightness of two fireflies (dimmer one is attracted by brighter one) – brightness decreases as distances increase (brightest firefly will move randomly);
3. brightness of firefly is determined by value of objective function.

This has been successfully applied to solve many real-world optimization problems in different domains such as the traveling salesman problem, time series analysis, and so on [32, 38]. Cluster analysis based on meta-heuristics (and particularly with the firefly algorithm) has been a hot research topic in recent years since it does not need preconditions to cluster (which is a requirement of traditional clustering algorithms). An experimental analysis and thorough literature survey have proven that the proposed methodology with the firefly algorithm offers better performance on clustering results [4, 22, 32, 36, 39].

In the present work, we address the clustering of benchmark data sets using firefly algorithm-based clustering algorithms. We propose three different fitness functions and evaluate the performance. Finally, we employed the proposed algorithms to find the playing strategies of a set of badminton players. For the convenience of the reader, the rest of the article is organized in the following manner. Section 2 provides basic background information about nature-inspired algorithms' involvements in solving the problem of data clustering. Section 3 provides information about the materials and methods that were used in the study. In Section 4, the proposed fitness functions of the firefly algorithm are completely explained. Section 5 is dedicated to the results. Finally, in Section 6, we conclude the findings and discuss future possibilities.

2. Related works

Much research that is related to cluster analysis can be found in the literature, as it is a trending topic and can be applied in various fields such as bio-informatics, web analysis, image analysis, and many others. However, only a limited amount of research is available on clustering with nature-inspired optimization algorithms. Here, we try to summarize most of this work.

Akay et al. recommended the use of a genetic algorithm with a new fitness function to solve the clustering problem [3]. The aim was to provide the optimal clustering of units by using the genetic algorithm. They generated a new population using genetic operators and worked on the chromosomes. The proposed GA was applied to artificial data sets, and the results have been compared with the k-means and Ward's methods [8, 14]. Their results showed that the recommended GA with the new fitness function was better and more powerful than Ward's method and the k-means algorithm.

The authors of "Finding Roles of Players in Football Using Automatic Practice Swarm Optimization (PSO) Clustering Algorithm" focused on the clustering of a football data set in order to find the different roles of players in a match by using an automatic PSO algorithm [7]. In this work, an automatic big data-clustering method that was based on a swarm-intelligence algorithm was proposed to automatically cluster a data set of players' performance centers in different matches and extract different kinds of roles in football. Their method was tested on six synthetic data sets, and the performance was compared with two other conventional clustering methods. According to the results of the experiment, the authors demonstrated that the different

roles of football players in a match based on their positions on a field can be used for different tasks (such as player rankings), and the automatic PSO was successful in finding better cluster centroids.

One of the major areas in clustering is document clustering. Text document clustering refers to the clustering of related text documents into groups based upon their content. This is a fundamental operation that is used in unsupervised document organization, text data mining, automatic topic extraction, and information retrieval [18]. Due to the considerable disadvantages of conventional algorithms such as k-means, nature-inspired optimization algorithms have been recognized as efficient substitutes for them. Several studies can be found where particle swarm optimization is hybridized with the k-means and fuzzy c-means algorithms to cluster document data sets [10, 18]. The results indicated that the hybrid PSO algorithm provided more-accurate clustering results than traditional clustering techniques alone.

Many meta-heuristics (including evolutionary and swarm-intelligence algorithms) were effectively used for the purpose of image clustering. Particle swarm optimization and the bee, firefly, and genetic algorithms were applied to solve many image-clustering tasks [13, 17, 23, 31, 35].

The latest research work has discovered the interesting subdivision capability of the firefly algorithm; hence, several studies have been conducted using the firefly algorithm (FA) as the main clustering technique with different modifications. Here, we summarize a few of these attempts.

Senthilnath et al. carried out work whose findings revealed that the capability of the firefly algorithm was greater than other nature-inspired algorithms in identifying the best cluster centroids when compared regarding the performance of solving 13 benchmark clustering problems [32]. Furthermore, [30] optimized energy consumption in wireless sensor networks by using FA while clustering the sensor nodes into small groups; this prolonged the lifetime of the network over the PSO and LEACH (low-energy adaptive clustering hierarchy [16]) methods that were used in previous works. In [21], FA selected cluster heads in a wireless sensor network (WSN) while working as an aggregator to reduce the duplicate data that was transmitted by the sensor nodes; this was done more efficiently than with the LEACH [16] and SFA [1] models.

Moving further along with the hybrid concept, several studies have been conducted to check the performance of hybrid k-means/firefly algorithms. In [15], the authors used a hybrid clustering algorithm that cooperated with the k-means algorithm and FA, which randomly provided the initial guesses to overcome the initial sensibility problem of the k-means algorithm and the obtained clusters with a minimum number of errors. The authors in [33] implemented a firefly variant with the k-means algorithm to gain the best centroids without becoming trapped in the local optima to solve the problem of image segmentation. In [19], the authors presented a forecasting application with an FA-based k-means algorithm for cluster analysis as well as an FA-based SVR for developing a forecasting model for each cluster. Having avoided the famous issue in k-means on initialization, [22] proposed a hybrid solution

for k-means with FA so that the most suitable solution for the local optimum could be obtained. Addressing the drawbacks of both [15] and [22], the authors of [39] proposed two variants of hybrid FA/k-means algorithms to improve the limitations of previous hybrid versions. Furthermore, two variants of the firefly algorithm were implemented in [36] in order to reduce the k-means clustering algorithm's existing drawbacks while improving the algorithm's efficiency.

There have been a few studies that have focused on combining FA with evolutionary algorithms to address clustering tasks. Using an FA-based improved genetic algorithm, Kaushik and the team presented a hybrid algorithm [20] whose performance was better when compared to canonical FA and canonical GA separately when applied to four benchmark data sets: Iris, Glass, Brest cancer, and Wine (taken from the UCI repository [11]). Another combination of FA and genetic algorithms was applied to select the appropriate cluster heads of a wireless sensor network [6]. Via this combination, significant minimizations of the packet loss and the end-to-end delay were obtained. Furthermore, it was able to increase the number of clusters to reduce the energy consumption as compared to LEACH, EEFH, and the classical firefly algorithm. In addition, [28] proposed an improved firefly algorithm for data clustering coupled with a differential evolution whose performance was better than k-means and FA alone.

FA and PSO are considered to be similar meta-heuristics; therefore, their combination may not be of much interest. However, the literature features some clustering studies that employ this combination. The authors of [5] implemented a variant of FA that combined the concepts in PSO to solve the clustering problem of information retrieval from the web. The results showed that the fast convergence and the stability of the proposed FClust was higher than those of PSO and DE. Also, the authors of [2] used FA with PSO to reduce some drawbacks in the canonical firefly algorithm, such as the need for proper parameter tuning and the reduction of speed and convergence when the dimension of a problem grows. Furthermore, [26] proposed FA coupled with PSO (HFAPSO) to modify the LEACH-C algorithm in an effective manner to cluster head sections in WSNs.

Although hybrid algorithms appear to be more effective, drawbacks such as their increased time and memory consumption can lower their usability. Since the canonical firefly algorithm has inborn capabilities of sub-swarming, we were interested in observing FA along with different fitness measures. Therefore, the present study focuses on applying different fitness evaluations in order to find the clustering improvements of the firefly algorithm.

3. Materials and methods

3.1. Firefly algorithm (FA)

The basic concept of FA is that each solution for a selected problem is considered to be (and called) a firefly. All of the fireflies have a light intensity (I) or a fitness value.

Light intensity (I) is associated with the objective function or fitness function $f(x)$. During the iterations, the fireflies in a population will move toward the better fireflies.

Algorithm 1 : Pseudo code of firefly algorithm

```
Begin;
Initialize algorithm parameters:
   MaxGen: maximum number of generations
   γ : light absorption coefficient
   β0 : initial brightness of firefly
   d : domain space
Define objective function f(X), where X = (x1, ..., xd)^T
Generate initial population of fireflies, X_i,   (i = 1, 2, ..., n)
Determine light intensity of I_i at i^th firefly X_i via f(X_i)
while t < MaxGen do
   for i = 1 : n (all n fireflies) do
      for j = 1 : n (all n fireflies) do
         if I_j > I_i then
            Move firefly i toward j by using Equation (1);
         end if
         Attractiveness varies with distance r via e^(-γr^2) using Equation (2);
         Evaluate new solutions and update light intensity;
      end for
   end for
   Rank fireflies and find current best;
end while
Post process results and visualization;
```

In FA, the parameter set should be properly specified. After the initial steps, the fireflies in the population start moving toward brighter fireflies according to the following equation:

$$x_i^{t+1} = x_i^t + \beta(x_j^t - x_i^t) + \alpha(rand - 0.5) \tag{1}$$

$$\beta = \beta_0 e_{ij}^{-\gamma r^2} \tag{2}$$

where α is a scaling factor (controlling the step sizes of the random walks) and γ is a scale-dependent parameter (controlling the visibility of the fireflies). In addition, β_0 is the attraction at $r = 0$, where r is the distance between two fireflies. *rand* is a random number that is generated from a uniform distribution.

The three terms in Eq. (1) represent the contribution from a current firefly, the attraction between two fireflies, and a randomization term, respectively. The equation supports both exploitation and exploration. Parameter α plays an important role in the randomization process, which is from uniform or Gaussian distribution. Over the

iterations, α should be gradually decreased in order to reduce exploration over time. For this, the value of α is decreased by δ amount in each iteration:

$$\alpha = \alpha.\delta \tag{3}$$

where $\delta \in [0, 1]$.

In the original implementation, Yang proposed some initializations as $\beta_0 : 1$, $\gamma \in [0.01, 100]$ and $\delta \in [0.9, 1]$.

To control the randomness, Yang used the randomness reduction factor after each iteration; this reduces according to Eq. (3). Considering the performance of this algorithm for cluster analysis, we are motivated to see the captivating behavior of the firefly algorithm in the process of clustering.

3.2. Benchmark data sets

The proposed algorithm has been tested with several benchmark synthetic data sets. The synthetic **S** set has four data sets (**S1, S2, S3, S4**), which consist of synthetic 2-D data with N = 5000 vector and k = 15 Gaussian clusters with different degrees of cluster overlap [12]. So, k is selected to be 15 for this study. Figure 1 indicates the distribution of the four data sets.

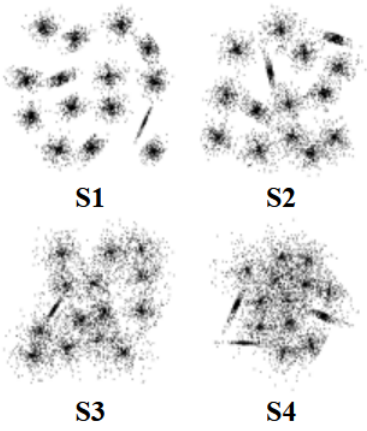

Figure 1. Distribution of S data sets

Similarly, synthetic set **A** has three data sets (**A1, A2, A3**), which consist of synthetic 2-D data with different sizes and an increasing number of clusters (k = 20, 35, 50).

Considering higher-dimensional data, two DIM sets were selected (**dim032** and **dim064**) with dimensions of 32 and 64, respectively. Each data set consists of 1024 data points and k = 16 clusters.

More information on these benchmark sets can be obtained via https://cs.joensuu.fi/sipu/datasets/.

One medical data set was also used to test the algorithms. The data set had 3201 records with 21 features such as the visual acuity of right and left eyes, contrast sensitivity, color perception, visual integration, choroidal thicknesses, etc. The data was collected from patients with Alzheimer's disease (AD). Patients were categorized into three stages (three clusters): mild AD, moderate AD, and control healthy persons [29].

Apart from the benchmark sets and the medical data set, we used the proposed algorithm variants on two data sets as an application that contained 41 badminton players (21 females/20 males) that featured ten different body measurements: height, weight, arm length, leg length, upper arm girth, forearm girth, thigh girth, calf girth, ankle girth, and body fat. The goal was to separate the players into two main playing styles (singles, and doubles).

3.3. Cluster validation indices

There are various clustering-validation indices that are available to measure the quality of a clustering, including the Rand index, adjusted Rand index, normalized mutual information index, etc. We tested our algorithm on four synthetic data sets (S1, S2, S3, S4), and its accuracy was calculated by measuring the Rand index and the adjusted Rand index.

The Rand index (RI) measures the similarity and consistency between the resultant groups of two random clusterings of a data set [27]. The comparison is conducted for all data points in each group (cluster) of each clustering (partition). In other words, RI examines whether two data points are in the same cluster in all partitions or whether they are in different clusters between partitions. Given two data points (x, y), the two points (x, y) are paired points if they exist in the same cluster of a partition. Eq. (4) defines the formula of the Rand index given two partitions (P1, P2) and n data points.

$$\text{Rand Index (RI)} = \frac{a+d}{a+b+c+d} = \frac{a+d}{{}^{n}\mathrm{c}_2} \tag{4}$$

where a is the number of paired points in the two partitions (P1, P2). Parameter d is the number of points that are not paired in any of the partitions (P1, P2), b is the number of points that are paired in one partition (P1), and c is the number of points that are paired just in the second partition (P2). The mathematical formula for calculating the total number of pair that can be formed from a set of n elements is ${}^{n}\mathrm{c}_2$ and ${}^{n}\mathrm{c}_2 = n!/(2! \cdot (n-2)!)$.

The Rand index always takes on a value between 0 and 1, where 0 indicates that two clustering methods do not agree on the clustering of any pair of elements, and 1 indicates that two clustering methods perfectly agree on the clustering of each pair of elements. So, a higher Rand index with a value closer to 1 indicates that the data points are clustered very well.

The adjusted Rand index (ARI) is a widely used metrics for validating clustering performance. The Rand index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. The raw RI score is then "adjusted for chance" into the adjusted Rand index score by using the following scheme:

$$\text{ARI} = \frac{\text{RI} - \text{Expected value(RI)}}{Max(\text{RI}) - \text{Expected value(RI)}} \tag{5}$$

The adjusted Rand index is thus ensured to have a value close to 0 for random labeling independently of the number of clusters and samples and exactly 1 when the clusterings are identical (up to a permutation).

4. Firefly algorithm-based clustering algorithms

The foremost thing when solving a clustering problem along with the firefly algorithm is that it requires an appropriate encoding mechanism to represent the fireflies and designing a fitness function (objective function) for the problem. Our fitness functions (which are designed to solve a clustering problem) search the solution space to find the best possible centroids. To find the best set of centroids (in other words, to find the best fireflies – possible solution), a fitness function should be used to measure the quality of the fireflies. This function should measure the quality of the partitioning that is proposed by each firefly (that is, cluster centroids).

4.1. Population initialization

As the first step of the algorithm, a random population is generated. In this step, the fireflies should be positioned randomly in a solution space. For a better representation, the fireflies should contain the position of k centroids; this means that each firefly is represented as an array that contains k * p elements, where k is the number of clusters, and p is the number of features in the data set (or the dimension). In Figure 2, for example, a firefly with k centroids for a 2D data set is demonstrated. In this figure, C_{ij} is the j^{th} dimension of the i^{th} centroid.

Figure 2. Representation of firefly

The initial population consists of fireflies that represent centroids that are randomly selected from a data set instead of generating random numbers for the positions of the fireflies. Each firefly will carry k centroids. n fireflies are generated (where n is the population size).

4.2. New fitness functions

For the implementation of the algorithm, three new fitness functions were introduced. We used intra-cluster distance (distance within clusters), inter-cluster distance (distance between clusters), the silhouette value, and the Calinski Harabasz value to compute the fitness functions.

Intra-cluster distance (or the within distance) is associated with clustering the similarity of points within a single cluster. The objective of a clustering problem is to maximize the similarity among the data points. The closer the points are, the more similar they are. So, the intra-cluster distance is considered to be minimized. Mainly, three criteria can be used to calculate the intra-cluster distance: the complete diameter, the average diameter, and the centroid diameter. In the complete diameter, the intra-cluster distance is characterized by the length of the link between the furthest two points in the cluster. For the average diameter, the distance is represented by the average of the distances among all of the data points in the cluster. Meanwhile, the centroid diameter is calculated by the average distance between all of the data points and the centroid.

In our work, the within distance in the k^{th} cluster is calculated using the complete diameter distance; the distances are drawn from Euclidean measurements (see Eq. (6)). Here, n indicates the number of fireflies:

$$\text{Within Distance (WD)} = \sum_{i=1}^{n} \sum_{j=1}^{n} (X_i - X_j)^2 \tag{6}$$

Then, a summation of the within distances is calculated for each firefly according to Eq. (7). Here, i gives the index of the summation, and k is the upper limit of the summation (which is the number of clusters):

$$\text{Sum (WD)} = \sum_{i=1}^{k} \text{WD} \tag{7}$$

The inter-cluster distance (or the between distance) shows the dissimilarity of the clusters. When the clusters are more separated from each other, they are more dissimilar. The inter-cluster distance is measured by using the following Eq. (8):

$$\text{Between Distance (BD)} = \sqrt{\sum_{i=1}^{q} \sum_{j=1}^{r} (X_i - X_j)^2} \tag{8}$$

where q and r are the number of elements in two clusters. Then, a summation of the between distances is calculated for each firefly.

The silhouette value is a measure of how similar an object is to its own cluster as compared to other clusters (see Eq. (9)). The value of the silhouette ranges between $[1, -1]$, where a high value indicates that an object is well-matched to its own cluster and poorly matched to its neighboring clusters:

$$\text{Silhouette Value} = \frac{b_i - a_i}{max(b_i - a_i)} \tag{9}$$

where b_i is the average distance from the i^{th} point to the points in another cluster, and a_i is the average distance from the i^{th} point to the other points in its own cluster.

The Calinski-Harabasz (CH) index is an object that consists of sample data, clustering data, and Calinski-Harabasz criterion values that are used to evaluate the optimal number of clusters. The Calinski-Harabasz criterion is sometimes called the variance ratio criterion (VRC); this is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters (the higher the score, the better the performance). Well-defined clusters have a large between-cluster variance and a small within-cluster variance. The optimal number of clusters corresponds to the solution with the highest Calinski-Harabasz index value. The CH index for K clusters on a data set $D = [d_1, d_2, d_3, ..., d_N]$ is defined as is shown in Eq. (10):

$$\text{Calinski-Harabasz Value} = \frac{\frac{\sum_{k=1}^{K} n_k \left\| c_k - c \right\|^2}{k-1}}{\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| d_i - c_k \right\|^2}{N-K}} \tag{10}$$

where n_k and c_k are the numbers of points and centroids of the k^{th} cluster, respectively, c is the global centroid, and N is the total number of data points.

The proposed firefly algorithm aims to maximize the objective functions, which can be expressed as follows:

$$\text{Fitness Function 1} = \frac{\text{BD}}{\text{Sum(WD)}} + \text{Silhouette Value} \tag{11}$$

$$\text{Fitness Function 2} = \frac{\text{BD}}{\text{Silhouette Value}} \tag{12}$$

$$\text{Fitness Function 3} = \frac{1}{\text{Calinski} - \text{Harabasz Value}} \tag{13}$$

5. Results

The experiments that were performed in this study are detailed here. The data sets that were used to verify the results are listed in Section 3.2. A summary of the data sets is given in Tables 1, 2, and 3. For the three benchmark data sets (S, A, and DIM) the ground truth centroids and partitions can be obtained from https://cs.joensuu.fi/sipu/datasets/, making comparison tasks easier.

All of the work in this research was carried out on an Intel Core i3 laptop with 2.30 GHz and 2 GB of RAM. MATLAB was used as the programming language. The initial parameter values of the firefly algorithm were set as $\alpha = 0.2$, $\gamma = 1$, $\delta = 0.94$, and $\beta_0 = 1$. Moreover, the size of the population was considered to be a variable parameter (most of the time, this was set at 10, 20, or 50).

Table 1
Characteristics of synthetic S data sets

	Data set	Data points	Number of features	Number of clusters
S data set	S1	5000	2	15
	S2			
	S3			
	S4			

Table 2
Characteristics of synthetic A data sets

	Data set	Data points	Number of features	Number of clusters
A Data set	A1	3000	2	20
	A2	5250		35
	A3	7500		50

Table 3
Characteristics of synthetic DIM data sets

	Data set	Data points	Number of features	Number of clusters
DIM-sets (high)	dim032	1024	32	16
	dim064		64	

Cluster validity index values were calculated for all nine benchmark data sets for the three firefly variants with different fitness functions. The fitness functions that are defined in Equations (11), (12), and (13) were used for the fitness calculations and evaluations of the algorithms. The obtained results are given in Tables 4, 5, 6, 7, 8, 9, 10 and in Figure 3.

Table 4 shows the performance (the mean RI and mean ARI values) for the S1 data set with different fitness functions (F1, F2, F3) for different population sizes (10, 20, 50) and different iterations of the algorithms. The mean values were calculated for 20 runs of each firefly variant. For example, the value of 0.9305 in the first row of Table 4 gives the mean RI value for the S1 data set for fitness function F1 when the algorithm runs with 10 fireflies and 20 iterations for 20 runs.

As shown below, Fitness Function 3 showed the highest score for the mean Rand index as well as the mean adjusted Rand index for the S1, S2, and S4 data sets. Out of the two metrics, the Rand index showed the highest value when comparing the performances of the four data sets.

Similarly, higher RI values were given by all of the fitness functions (particularly, Fitness Function 3) for the other synthetic data sets and the medical data set.

Table 4
Mean RI and mean ARI values obtained for S1 data set over 20 runs

	Fitness Function	Fireflies	Rand Index (RI) / Adjusted Rand Index (ARI)	Iterations		
				20	50	100
S1	F1	10	RI	0.9305	**0.9432**	0.9373
			ARI	0.5270	**0.6347**	0.6046
		20	RI	0.9311	0.9386	**0.9423**
			ARI	0.5603	0.6014	**0.6270**
		50	RI	0.8900	0.8678	**0.8930**
			ARI	0.3884	0.3475	**0.3991**
	F2	10	RI	**0.9762**	**0.9762**	0.9755
			ARI	**0.8196**	**0.8196**	0.8148
		20	RI	0.9759	0.9758	**0.9763**
			ARI	0.8179	0.8178	**0.8223**
		50	RI	**0.9671**	0.9635	0.9648
			ARI	**0.7625**	0.7420	0.7452
	F3	10	RI	**0.9762**	0.9760	0.9760
			ARI	**0.8217**	0.8192	0.8186
		20	RI	0.9759	**0.9763**	0.9760
			ARI	0.8185	**0.8216**	0.8184
		50	RI	**0.9818**	0.9765	0.9765
			ARI	**0.8592**	0.8210	0.8227

Table 5
Mean RI and mean ARI values obtained for S2 data set over 20 runs

	Fitness Function	Fireflies	Rand Index (RI) / Adjusted Rand Index (ARI)	Iterations		
				20	50	100
S2	F1	10	RI	0.9385	**0.9409**	0.9322
			ARI	0.5604	**0.5628**	0.5193
		20	RI	**0.9207**	0.9082	0.9049
			ARI	**0.5038**	0.4434	0.4262
		50	RI	0.8891	**0.9050**	0.8868
			ARI	0.4042	**0.4659**	0.3989
	F2	10	RI	0.9696	0.9706	**0.9751**
			ARI	0.7688	0.7768	**0.8112**
		20	RI	**0.9704**	0.9484	0.9631
			ARI	**0.7757**	0.6389	0.7119
		50	RI	0.9517	0.9542	**0.9699**
			ARI	0.6695	0.6781	**0.7750**
	F3	10	RI	0.9475	**0.9475**	0.9474
			ARI	0.5920	0.5942	**0.5959**
		20	RI	**0.9843**	0.9412	0.9406
			ARI	**0.6752**	0.5509	0.5522
		50	RI	**0.9499**	0.9497	0.9483
			ARI	0.6062	**0.6121**	0.6054

The purpose of the performed experiments was to evaluate the performance of the proposed algorithms on different data sets; we used data sets with different characteristics.

The results indicated that the proposed FA-based clustering algorithms worked well on data sets with high numbers of clusters. With these successful findings, the algorithms were then tested with real data sets of badminton players in order to identify their playing strategies.

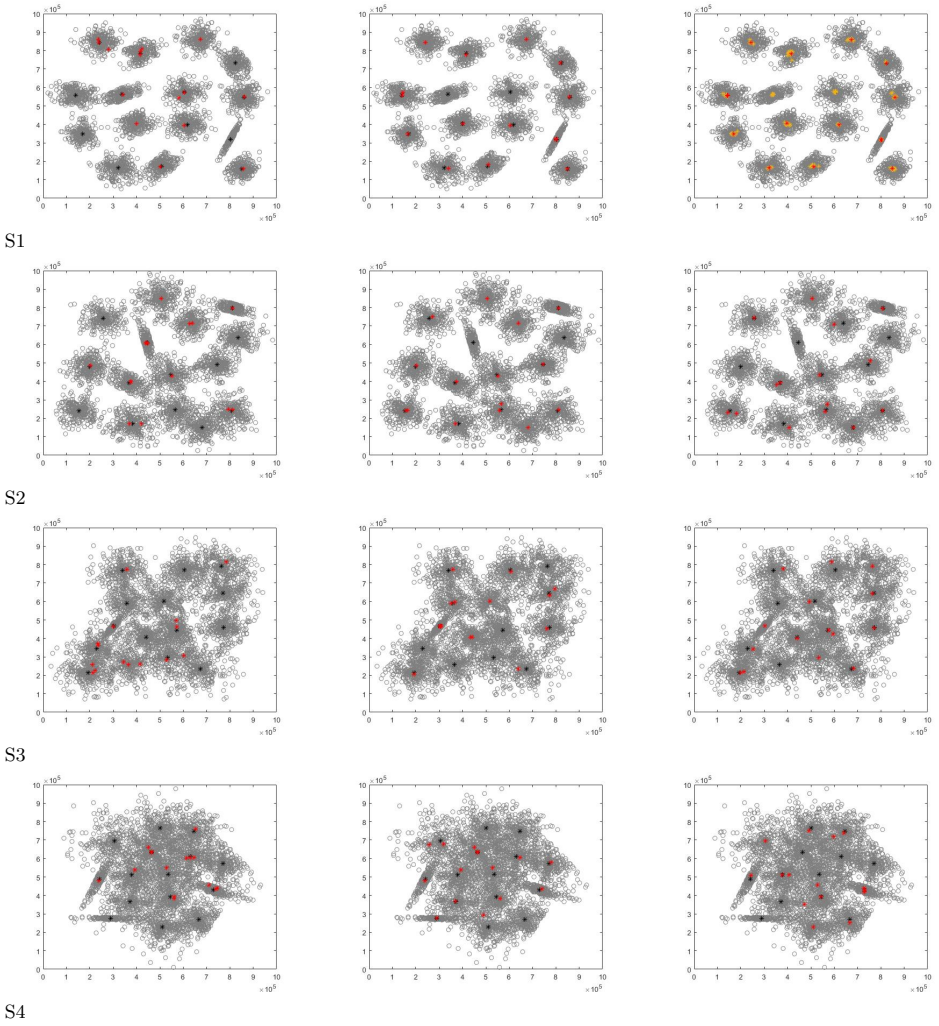

Figure 3. Centroids of S data sets found by FA-based clustering algorithms

Table 6
Mean RI and mean ARI values obtained for S3 data set over 20 runs

	Fitness Function	Fireflies	Rand Index (RI) / Adjusted Rand Index (ARI)	Iterations		
				20	50	100
S3	F1	10	RI	**0.9116**	0.9078	0.8981
			ARI	**0.4309**	0.4245	0.3979
		20	RI	**0.9232**	0.9146	0.8968
			ARI	**0.4942**	0.4567	0.4173
		50	RI	0.9049	**0.9110**	0.9038
			ARI	0.4376	**0.4550**	0.4251
	F2	10	RI	0.9350	0.9212	**0.9367**
			ARI	0.5238	0.4854	**0.543**
		20	RI	0.9200	0.9319	**0.9409**
			ARI	0.4821	0.5285	**0.5613**
		50	RI	0.9345	**0.9570**	0.9454
			ARI	0.5500	**0.6645**	0.5895
	F3	10	RI	0.9305	0.9337	**0.9358**
			ARI	0.4767	0.4987	**0.5162**
		20	RI	0.9364	0.9349	**0.9366**
			ARI	0.5324	0.5180	**0.5358**
		50	RI	0.9349	0.9349	**0.9413**
			ARI	0.5236	0.5012	**0.5465**

Table 7
Mean RI and mean ARI values obtained for S4 data set over 20 runs

	Fitness Function	Fireflies	Rand Index (RI) / Adjusted Rand Index (ARI)	Iterations		
				20	50	100
S4	F1	10	RI	0.8988	0.9041	**0.9062**
			ARI	0.3542	0.3553	**0.3768**
		20	RI	0.9035	**0.9179**	0.9175
			ARI	0.3847	**0.4308**	0.4286
		50	RI	0.9067	**0.9170**	0.9124
			ARI	0.3725	**0.4214**	0.3946
	F2	10	RI	0.9355	0.9344	**0.9367**
			ARI	0.5241	0.5093	**0.5323**
		20	RI	0.9298	0.9358	**0.9417**
			ARI	0.4833	0.5282	**0.5513**
		50	RI	0.9356	**0.9360**	0.9323
			ARI	0.5042	**0.5236**	0.5059
	F3	10	RI	0.9305	0.9337	**0.9358**
			ARI	0.4767	0.4987	**0.5162**
		20	RI	0.9364	0.9349	**0.9366**
			ARI	0.5324	0.5180	**0.5358**
		50	RI	0.9349	0.9349	**0.9413**
			ARI	0.5236	0.5012	**0.5465**

Table 8
Mean RI and mean ARI values obtained for A1, A2, and A3 data sets over 20 runs (each with 50 iterations)

	Fitness function	Mean RI	Mean ARI
A1	F1	0.93902	0.50735
	F2	0.95414	0.60691
	F3	0.96722	0.68059
A2	F1	0.97130	0.59648
	F2	0.98194	0.70319
	F3	0.97680	0.63879
A3	F1	0.98650	0.68899
	F2	0.98471	0.66265
	F3	0.98550	0.67639

Table 9
Mean RI and mean ARI values obtained for Dim032 and Dim064 data sets over 20 runs (each with 50 iterations)

Dim data sets	Fitness function	Mean RI	Mean ARI
Dim032	F1	0.9667	0.7112
	F2	0.9741	0.7564
	F3	0.9843	0.7676
Dim064	F1	0.9693	0.7535
	F2	0.9899	0.7645
	F3	0.9979	0.7756

Table 10
Mean RI and mean ARI values obtained for Alzheimer data set over 20 runs (each with 50 iterations)

	Fitness function	Mean RI	Mean ARI
Alzheimer Data Set	F1	0.9534	0.5467
	F2	0.9728	0.6732
	F3	0.9821	0.7321

5.1. Comparisons with other algorithms

In order to evaluate the performances of the proposed algorithms, we compared them with k-means [14], x-means [24], and APSO-clustering [7] algorithms for the S1, S2, A1, A2, and Dim064 data sets (Tab. 11). The comparisons were restricted to some extent since we did not regenerate the algorithms but used the results that were provided in the original paper [7].

Table 11
Comparisons with APSO, k-means, and x-means

Method	Rand Index				
	S1	S2	A1	A2	Dim064
FF Algorithm Based Clustering F1	0.9432	0.9409	0.9390	0.9713	0.9693
FF Algorithm Based Clustering F2	0.9763	0.9704	0.9541	0.9819	0.9899
FF Algorithm Based Clustering F3	0.9818	0.9843	0.9872	0.9968	0.9979
APSO-Clustering	0.9959	0.9911	0.9981	0.9975	0.9990
k-means	0.9901	0.9777	0.9877	0.9924	0.9984
x-means	0.9225	0.9353	0.8620	0.9104	0.9844

The results indicated that the proposed algorithms equally performed with the clustering method that was implemented using the other popular meta-heuristic (APSO) and surpassed the performances of the traditional clustering algorithms.

Since the proposed clustering algorithms competed well, we applied the algorithms to cluster two data sets that included university badminton player details. The aim was to identify the playing styles of the players based on their physical characteristics.

5.2. Description of badminton data

Badminton is a racket sport that is played using a racket to hit a shuttlecock across a net. Although it can be played with larger teams, the most common forms of the game are "singles" (with one player per side) and "doubles" (with two players per side) [25]. These playing styles are primarily based on the individual player's strengths and preferences. We used two data sets that contained the following 10 features for 21 female players and 20 male players (see Tab. 12).

- height,
- weight,
- arm length,
- leg length,
- upper arm girth,
- forearm girth,
- thigh girth,
- calf girth,
- ankle girth,
- body fat.

The goal was to divide the data sets with similar characteristics into two groups. We propose that an expert coach can identify singles and doubles players by considering these characteristics. Then, when an amateur/new player enters the ring, the coaches can easily identify the best playing style and advise them on such selections. We employed three firefly variants that were proposed in the study on male and female data sets separately. After obtaining the two clusters, we named the clusters as singles and doubles player clusters by looking at the manual separation. The silhouette plots and the mean silhouette values were calculated for the clustering results that were obtained for both the male and female data sets.

Table 12
Characteristics of badminton data sets

	Number of data points	Number of features	Number of Clusters
Badminton Data set	41 (21 Females/20 males)	10	2 (Single/Double)

The silhouette plot displays a measure of how close each point in one cluster is to the points in its neighboring clusters; therefore, it provides a way to assess parameters such as the suitability of the number of clusters in a visual manner. Silhouette coefficients can be obtained for each and every data point; silhouette coefficients that are near +1 indicate that the sample is far away from their neighboring clusters. A value of 0 indicates that the sample is on (or very close to) the decision boundary between two neighboring clusters, and negative values indicate that those samples might have been assigned to the wrong cluster.

We calculated the silhouette scores for each data point in the male data set and obtained the mean silhouette score of the data set. This was done with all three fitness functions/FF variants; the mean silhouette scores that were obtained for the male data were 0.46631, 0.46197, and 0.46197. The mean values indicated that the selected number of clusters ($k = 2$) was acceptable since the presence of the clusters were greater than the average silhouette scores. For the second and third variants of the algorithm (fitness functions), all of the silhouette scores were positive, which gave the idea that the sample/record was away from its neighboring clusters (Fig. 4 and Tab. 13).

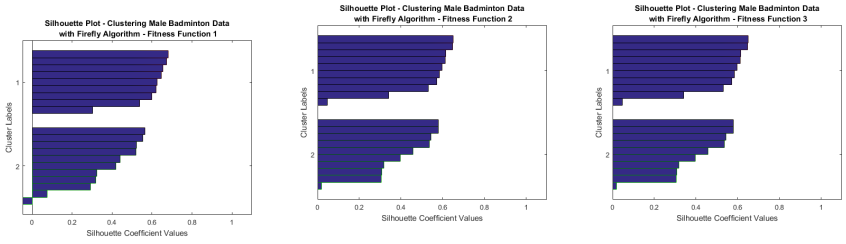

Figure 4. Silhouette plots for three fitness functions – male badminton data

Table 13
Mean silhouette score – badminton data (male)

	Fitness function/FFvariant	Mean silhouette score
Badminton Data Male	F1	0.46631
	F2	0.46197
	F3	0.46197

For the female data set, the first two fitness functions behaved similarly, giving an average silhouette score of 0.27714, while the third fitness function gave an average silhouette score of 0.30771. These average silhouette scores indicated that the number of clusters ($k = 2$) was a good choice (Fig. 5 and Tab. 14). However, the negative silhouette scores that were achieved for several individual player records indicated the wrong clustering of such players/outliers. When compared with the male data set, the quality of the clustering was not better in the female data set. The reason for this may have been the nature of the data, which can be explained by the fact that most of these female players could play both singles and doubles playing styles.

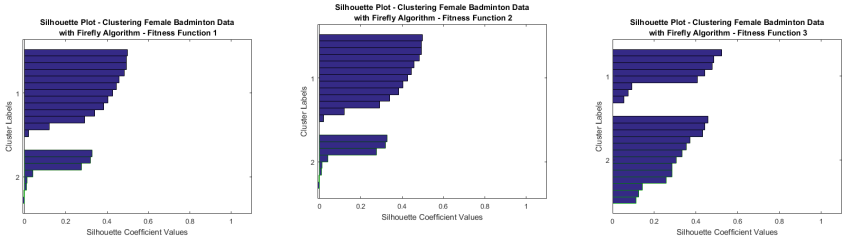

Figure 5. Silhouette plots for three fitness functions – female badminton data

Table 14
Mean silhouette score – badminton data (female)

	Fitness function/FFvariant	Mean silhouette score
Badminton Data Female	F1	0.27714
	F2	0.27714
	F3	0.30771

6. Conclusions

The proposed algorithms (more preciously, the FF variants) that were formed by converting a clustering problem to an optimization problem were successfully implemented on the data-clustering problem. The results that are presented here from various benchmark problems indicated that the proposed firefly variants with different fitness functions were successful in finding the suitable cluster centroids for the

given data sets. The proposed algorithms surpassed the difficulties in standard clustering techniques such as the dependability of the initial centroids and being resistant to high-dimensional problems. For the purpose of comparison, we used methods that were discussed in the literature (including a variant PSO algorithm). The obtained results illustrated that the firefly variants were equally capable of solving clustering problems as were the other meta-heuristics in the literature. As an application, the firefly variants were used to cluster two badminton data sets (male and female) for finding the different playing styles of players in badminton. The algorithms divided the data sets into two distinct clusters, indicating two main playing styles for the players. However, the clustering of the female data set gave low silhouette scores; this indicated that the data was closer to the decision boundaries (it can be concluded that most of the female players in the given data set had physical features that were similar and could be practiced in both playing styles). For future work, it is worth concentrating on the applicabilities of meta-heuristics to find the number of clusters (k) automatically.

References

[1] Abirami T., Anandamurugan S.: Data aggregation in wireless sensor network using shuffled frog algorithm, *Wireless Personal Communications*, vol. 90(2), pp. 537–549, 2016.

[2] Agbaje M.B., Ezugwu A.E., Els R.: Automatic data clustering using hybrid firefly particle swarm optimization algorithm, *IEEE Access*, vol. 7, pp. 184963–184984, 2019.

[3] Akay Ö., Tekeli E., Yüksel G.: Genetic Algorithm with New Fitness Function for Clustering, *Iranian Journal of Science and Technology, Transactions A: Science*, vol. 44(3), pp. 865–874, 2020.

[4] Ariyaratne M., Fernando T.: A Comprehensive Review of the Firefly Algorithms for Data Clustering, *Advances in Swarm Intelligence*, pp. 217–239, 2023.

[5] Banati H., Bajaj M.: Performance analysis of firefly algorithm for data clustering, *International Journal of Swarm Intelligence*, vol. 1(1), pp. 19–35, 2013.

[6] Baskaran M., Sadagopan C.: Synchronous firefly algorithm for cluster head selection in WSN, *The Scienti c World Journal*, vol. 2015, 2015.

[7] Behravan I., Zahiri S.H., Razavi S.M., Trasarti R.: Finding roles of players in football using automatic particle swarm optimization-clustering algorithm, *Big Data*, vol. 7(1), pp. 35–56, 2019.

[8] Blashfield R.K.: The growth of cluster analysis: Tryon, Ward, and Johnson, *Multivariate Behavioral Research*, vol. 15(4), pp. 439–458, 1980.

[9] Brucker P.: On the Complexity of Clustering Problems. In: R. Henn, B. Korte, W. Oettli (eds.), *Optimization and Operations Research*, pp. 45–54, Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[10] Cui X., Potok T.E., Palathingal P.: Document clustering using particle swarm optimization. In: *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pp. 185–191, IEEE, 2005.

[11] Dua D., Graff C.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, 2019.

[12] Fränti P., Virmajoki O.: Iterative shrinking method for clustering problems, *Pattern Recognition*, vol. 39(5), pp. 761–765, 2006. doi: 10.1016/j.patcog.2005.09.012.

[13] Hancer E., Ozturk C., Karaboga D.: Artificial bee colony based image clustering method. In: *2012 IEEE Congress on Evolutionary Computation*, pp. 1–5, 2012.

[14] Hartigan J.A., Wong M.A.: Algorithm AS 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society Series C (Applied Statistics)*, vol. 28(1), pp. 100–108, 1979.

[15] Hassanzadeh T., Meybodi M.R.: A new hybrid approach for data clustering using firefly algorithm and K-means. In: *The 16th CSI International Symposium on Arti cial Intelligence and Signal Processing (AISP 2012)*, pp. 007–011, 2012.

[16] Heinzelman W.R., Chandrakasan A., Balakrishnan H.: Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.

[17] Hrosik R.C., Tuba E., Dolicanin E., Jovanovic R., Tuba M.: Brain image segmentation based on firefly algorithm combined with k-means clustering, *Studies in Informatics and Control*, vol. 28(2), pp. 167–176, 2019.

[18] Karol S., Mangat V.: Evaluation of text document clustering approach based on particle swarm optimization, *Central European Journal of Computer Science*, vol. 3(2), pp. 69–90, 2013.

[19] Kuo R., Li P.: Taiwanese export trade forecasting using firefly algorithm based *K*-means algorithm and SVR with wavelet transform, *Computers* & *Industrial Engineering*, vol. 99, pp. 153–161, 2016.

[20] Maheshwar, Kaushik K., Arora V.: A Hybrid Data Clustering Using Firefly Algorithm Based Improved Genetic Algorithm, *Procedia Computer Science*, vol. 58, pp. 249–256, 2015.

[21] Manshahia M.S., Dave M., Singh S.: Firefly algorithm based clustering technique for Wireless Sensor Networks. In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1273–1276, IEEE, 2016.

[22] Mizuno K., Takamatsu S., Shimoyama T., Nishihara S.: Fireflies can find groups for data clustering. In: *2016 IEEE International Conference on Industrial Technology (ICIT)*, pp. 746–751, IEEE, 2016.

[23] Omran M., Engelbrecht A.P., Salman A.: Particle swarm optimization method for image clustering, *International Journal of Pattern Recognition and Arti cial Intelligence*, vol. 19(03), pp. 297–321, 2005.

[24] Pelleg D., Moore A.W.: X-means: Extending k-means with Efficient Estimation of the Number of Clusters. In: *ICML'00: Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, 2000.

[25] Phomsoupha M., Laffaye G.: The science of badminton: game characteristics, anthropometry, physiology, visual fitness and biomechanics, *Sports Medicine*, vol. 45(4), pp. 473–495, 2015.

[26] Pitchaimanickam B., Murugaboopathi G.: A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks, *Neural Computing and Applications*, vol. 32(12), pp. 7709–7723, 2020.

[27] Rand W.M.: Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, vol. 66(336), pp. 846–850, 1971.

[28] Sadeghzadeh M.: Data Clustering Using Improved Fire Fly Algorithm. In: *Information Technology: New Generations*, pp. 801–809, Springer, 2016.

[29] Salobrar-García E., de Hoz R., Ramírez A.I., López-Cuenca I., Rojas P., Vazirani R., Amarante C., *et al.*: Changes in visual function and retinal structure in the progression of Alzheimer's disease, *PloS one*, vol. 14(8), e0220535, 2019.

[30] Sarma N.V.S.N., Gopi M.: Implementation of energy efficient clustering using firefly algorithm in wireless sensor networks. In: *2014 1st International Congress on Computer, Electronics, Electrical, and Communication Engineering (ICCEECE2014)*, vol. 59, IACSIT Press, 2014.

[31] Scheunders P.: A genetic c-means clustering algorithm applied to color image quantization, *Pattern Recognition*, vol. 30(6), pp. 859–866, 1997.

[32] Senthilnath J., Omkar S.N., Mani V.: Clustering using firefly algorithm: performance study, *Swarm and Evolutionary Computation*, vol. 1(3), pp. 164–171, 2011.

[33] Sharma A., Sehgal S.: Image segmentation using firefly algorithm. In: *2016 International Conference on Information Technology (InCITe) { the Next Generation IT Summit on the Theme { Internet of Things: Connect Your Worlds*, pp. 99–102, IEEE, 2016.

[34] Welch W.J.: Algorithmic complexity: three *NP*-hard problems in computational statistics, *Journal of Statistical Computation and Simulation*, vol. 15(1), pp. 17–25, 1982.

[35] Wong M.T., He X., Yeh W.C.: Image clustering using particle swarm optimization. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 262–268, IEEE, 2011.

[36] Xie H., Zhang L., Lim C.P., Yu Y., Liu C., Liu H., Walters J.: Improving K-means clustering with enhanced firefly algorithms, *Applied Soft Computing*, vol. 84, 105763, 2019.

[37] Yang X.S.: Firefly algorithms for multimodal optimization. In: *International Symposium on Stochastic Algorithms*, pp. 169–178, Springer, 2009.

[38] Yang X.S., He X.: Firefly algorithm: recent advances and applications, *International Journal of Swarm Intelligence*, vol. 1(1), pp. 36–50, 2013.

[39] Zhou L., Li L.: Improvement of the Firefly-based K-means Clustering Algorithm. In: *Proceedings of the 2018 International Conference on Data Science*, pp. 157–162, 2018.

Affiliations

I.M.T.P.K. Ilankoon
University of Sri Jayewardenepura, Department of Computer Science, Faculty of Applied Sciences, Nugegoda, Sri Lanka, prabhaim95@gmail.com

U.S. Samarasinghe
University of Sri Jayewardenepura, Department of Computer Science, Faculty of Applied Sciences, Nugegoda, Sri Lanka, upekshasamarasingher@gmail.com

M.K.A. Ariyaratne
University of Sri Jayewardenepura, Department of Computer Science, Faculty of Applied Sciences, Nugegoda, Sri Lanka, mkanuradha@sjp.ac.lk

R.M. Silva
University of Sri Jayewardenepura, Department of Statistics, Faculty of Applied Sciences, Nugegoda, Sri Lanka, rsilva@sjp.ac.lk

Received: 24.11.2022
Revised: 18.02.2023
Accepted: 21.02.2023