Social Spider Algorithm Approach for Clustering

Harley Vera-Olivera* **José Luis Soncco-Álvarez*** **Lauro Enciso-Rodas***
harleyve@gmail.com, jose.soncco.alvarez@gmail.com, lauro.enciso@unsaac.edu.pe
National University of San Antonio Abad del Cusco*

Abstract

Clustering is a popular data analysis technique to identify homogeneous groups of objects based on the values of their attributes, used in many disciplines and applications. This extended abstract of our undergraduate thesis for obtaining the engineer degree in informatics and systems, presents an approach based on the Social Spider Optimization (SSO) algorithm for optimizing clusters of data, taking as metric the sum of euclidean distances. Other important algorithms of the literature were implemented in order to make comparisons: K-means algorithm, and a Genetic Algorithm (GA) for Clustering. Experiments were performed using 5 datasets taken from the UCI Machine Learning Repository, each algorithm was executed many times and then the following measures were calculated: mean, median, minimum, and maximum values of the results. These experiments showed that the SSO algorithm outperforms the K- means algorithm, and it has results equally competitive as the GA. All these results were confirmed by statistical tests performed over the outputs of the algorithm.

1 Introduction

Clustering is useful in several analysis such as exploration of patterns, machine learning including data mining, documents retrieval, image segmentation and pattern classification. However in many of these problems there is little prior information, such as statistical models. It is under these restrictions that clustering is particularly appropriate for the exploration of interrelationships between data to make a preliminary evaluation of its structure (Jain et al., 1999). Thus new conditions imposed by *Big Data* presented new challenges at different levels including clustering.

The term clustering is used in several communities to describe methods for grouping of unlabeled data (Jain et al., 1999). Clustering is the task of discovering groups and data structures that are in some way or another "similar", without using known structures (vijayalakshmi and Renuka Devi, 2012). Intuitively, patterns within a group are more similar compared to those patterns belonging to a different group. Here, the goal is to develop an automatic algorithm that can accurately classify an unlabeled dataset in groups.

Recent literature classifies clustering algorithms in hierarchical, partitioning, and overlapping (Xu and Wunsch, 2009). The partitional algorithm divides a dataset into a finite number based on certain criteria known as a measure of fitness. The fitness measure affects directly the natural formation of the groups, once a measure is selected the task of the partition becomes an optimization problem.

K-means algorithm is the most fundamental concept of partitional grouping, was published in 1957 by Lloyd (Lloyd, 1982). In this case the minimization of the Euclidean distance between its elements and the center of a cluster was considered as a criterion of optimization. Inspired by K-means many algorithms were developed such as: Bisecting K-means (Steinbach et al., 2000), sort-means (Phillips, 2002), X-means (Pelleg and Moore, 1999), among others.

Recent studies reveal a new trend, which was named as stochastic algorithms with randomized and local search meta-heuristic. The random pro-

This work was started when first and second authors were undergraduate students at the National University of San Antonio Abad del Cusco and was finished when the authors were graduate students at the University of Brasilia, both receiving a CAPES scholarship.

cess generates arbitrary solutions that explore the search space and are responsible for achieving global solution (Nanda and Panda, 2014). The first meta-heuristic inspired by nature was the genetic algorithm developed by Holland and his colleagues in 1975 (Holland, 1975). This algorithm is classified as evolutionary algorithm. On the other hand, new bio-inspired optimization algorithms are being introduced, such is the case of the algorithm inspired by the social behavior of spiders (Cuevas et al., 2013) classified as swarm intelligence algorithm proposed in 2013, which had not been applied to the clustering problem until our proposal.

This extended abstract of our undergraduate thesis for obtaining the engineer degree in informatics and systems (Vera-Olivera and Soncco-Álvarez, 2016), presents an approach based on the SSO algorithm for the clustering problem. The contribution of this work is to show that the SSO algorithm can produce competitive results regarding classic approaches such as: (a) the k-means algorithm, which was implemented as presented in (Maulik and Bandyopadhyay, 2000); and (b) a genetic algorithm approach for the clustering problem, which was proposed by Maulik and Bandyopadhyay (2000). The metric used for the comparisons is the sum of euclidean distances of the elements of the clusters to their respective center, this metric is the output of the algorithms. For the experiments were used 5 datasets from the UCI Machine Learning Repository, for each of these datasets the algorithms were executed several times, and then the following measures were calculated: mean, median, minimum, and maximum values. This experiment showed that the SSO algorithm has better results compared to the ones obtained by the K-means algorithm, also the SSO algorithm has equally competitive results as the GA. Additionally, a statistical analysis was performed, since we are working with stochastic algorithms, using the Kolmogorov-Smirnov test and the Wilcoxon rank sum test as discussed in (Demšar, 2006), (Durillo et al., 2009), (Muñoz et al., 2011). The results of these statistical test confirmed the results of the experiments.

This paper is organized as follow: in Section II, are given some definitions related to the clustering problem; in Section III is given the original proposal of the SSO algorithm; in Section IV, details of our approach based on the SSO algorithm for the clustering problem are presented, also the pseudo-code of the algorithms is presented; in Section V the experiments and results are showed, a discussion of this results is presented, and also a statistical analysis is performed; finally in Section VI are presented the conclusions and future work.

2 The Clustering Problem

According to Mirkin (Mirkin, 2012), clustering is a discipline dedicated to reveal and describe the structures of groups in a dataset and may define four important involved concepts: data, structure groups, reveal a group structure, and describe a group structure. The following definitions were taken from (Maulik and Bandyopadhyay, 2000; De Falco et al., 2007; Karaboga and Ozturk, 2011; Senthilnath et al., 2011).

Suppose $S = \{x_1, x_2, \dots x_n\}$ is a set of N-dimensional n points and $C = \{c_1, c_2, \dots, c_k\}$ is a set of N-dimensional k elements. The clustering problem in a N-dimensional space $\mathbb{R}^N$ consists in partitioning the set S in a number k of clusters based on a similarity metric, where each cluster has as center an element c_i from C.

Suppose that G_i, $i = 1, \dots, k$, represents a cluster, then the following properties hold:

- $G_i \neq \phi$, to $i = 1, \dots, k$;
- $G_i \cap G_j = \phi$, to $i, j = 1, \dots, k$, such that $i \neq j$;
- $\bigcup_{i=1}^{k} G_i = S$

The clustering metric that has been adopted in this work is the sum of the Euclidean distances of the points of a group to their respective center. The definition of this clustering metric $\mathcal{M}$ for k clusters $G_1, G_2, \dots, G_k$, is given by the following expression:

$$\mathcal{M}(G_1, G_2, \dots, G_k) = \sum_{i=1}^{k} \sum_{x_j \in G_i} \|x_j - c_i\|$$

3 Algorithm Based on the Social Behavior of Spiders

Cuevas et al. (2013) proposed a new optimization algorithm, called Social Spider Optimization(SSO), the development of this new algorithm

was guided by the operational principle of the social behavior of spiders. The SSO algorithm assumes that the solution space is a community network (spider web), where spiders interact to each others. The main features of this approach are:

- Each solution within a space of solutions represents the position of a spider in the community network.
- Each spider receives a weight according to the value of fitness solution that represents.
- The algorithm modeled two types of search agents (spiders): male and female. Depending on the genre each individual performs different types of operations that simulate their social behavior within the colony.

An important feature of the colonies of social spiders is that they have a high number of female agents. This fact is simulated by defining the number of females N_f randomly within the range of 65 to 90% of N, which is the number of elements of the total population. The number of males N_m is calculated as the complement of N_f regarding N.

The total population S is divided into two subgroups F and M. The group F is the set of female spiders, and the group M is the set of male spiders.

$$F = \{f_1, f_2, \ldots, f_{N_f}\}$$

$$M = \{m_1, m_2, \ldots, m_{N_m}\}$$

where $S = F \cup M = \{s_1, s_2, \ldots, s_N\}$

3.1 Calculation of Fitness

Each individual (spider) i of the population S receives a w_i weight, that represents the quality of its solution. This weight can be calculated as follows:

$$w_i = \frac{J(s_i) - worst_s}{best_s - worst_s}$$

where $J(s_i)$ is the fitness value calculated by evaluating the position of a spider s_i regarding the function J. The values $worst_s$ and $best_s$ considering a maximization problem, are defined as follows:

$$best_s = max(J(s_k)),\ k \in \{1, 2, \ldots, N\}$$

$$worst_s = min(J(s_k)),\ k \in \{1, 2, \ldots, N\}$$

3.2 Modeling of Vibrations Through the Community Network

The community network is used as a mechanism for transmitting information between the members of the colony. This information is coded as small vibrations that are critical for collective coordination of all individuals. The vibrations are based on the weight and the distance of the spider that generated it. The vibrations that are perceived by an individual i as a result of information transmitted by an individual j are modeled by the following expression:

$$Vib_{\ i,j} = w_j * e^{-d_{i,j}^2}$$

Where $d_{i,j}$ is the euclidean distance between spiders i and j. There are three special types of vibrations that are considered in the SSO algorithm:

- $Vib_{\ i,c}$ vibrations, where c is the closest member to i that has a higher weight compared to i ($w_c > w_i$).
- $Vib_{\ i,b}$ vibrations, where b is the individual who has the best weight (best fitness value) of the whole population S.
- $Vib_{\ i,f}$ vibrations, where f is the female individual closest to i.

3.3 Initialization of Population

The SSO algorithm starts by initializing the set S, which contains N spiders positions. Each position f_i or m_i, is an n-dimensional vector containing the values to be optimized. These values are distributed uniformly between the values, p^{low} and p^{high}, which are previously specified.

3.4 Cooperative Operators

3.4.1 Cooperative Operator for female spiders

To emulate the cooperative behavior of the female spiders, a new operator is defined. The operator considers the change in position of a female spider i at each iteration, this change can be attractive or repulsive and is calculated by combining three elements:

- The first element considers the change regarding the nearest member to i that has the highest weight and produces vibration $Vib_{\ i,c}$;

- The second element considers the change regarding the best individual of the population S that produces vibration $Vib_{i,b}$;
- The third incorporates a random movement.

The last three elements can be considered as one movement, we use the "+" symbol for attraction and the "-" symbol for repulsion. The change in position can be calculated as follows:

$$f_i^{k+1} = f_i^k \pm movement$$

where k represents the iteration number.

3.4.2 Cooperative Operator for male spiders

To emulate the cooperative behavior of the male spiders, these are divided into two groups: dominant D and non-dominant ND. This division is made according to its position respect to the $median$ of all male individuals. Individuals who have a weight that is above the $median$ are considered dominants, otherwise they are considered non-dominant.

For dominant males are defined two movements: (a) a movement of attraction to the nearest female f that produces a vibration $Vib_{i,f}$, and (b) a random movement. The last two movements can be considered as one, and then the change in a dominant male can be calculated as follows:

$$m_i^{k+1} = m_i^k + \text{D\_}movement$$

where k represents the iteration number.

For non-dominant males is defined just one movement of attraction to the weighted average of male spiders. Then the change in a non-dominant can be calculated as follows:

$$m_i^{k+1} = m_i^k + \text{ND\_}movement$$

where k represents the iteration number.

3.5 Mating operator

Mating a colony of spiders is made between females and dominant males. So when a dominant male m_g finds a set of female spiders E^g within a range of mating r, it mates, forming a new offspring S_{new}. This new offspring is generated from the set T^g, which is formed by the union of E^g and m_g. When the set T^g is empty, mating operation is canceled.

The weight of each spider that is involved in the mating process, i.e. spiders from the set T^g, defines a probability of influence on the new offspring. The probability of influence P_{s_i} is assigned using the roulette-wheel selection, which is defined as follows:

$$P_{s_i} = \frac{w_i}{\sum_{j \in T^g} w_j}$$

where $s_i \in T^g$.

A spider is a solution within the solution space, so a new spider is formed by choosing values for each variable, this variable is chosen within the values defined by the method of roulette. For example let $s_{new} = \{v_1, v_2, \ldots, v_n\}$ be the new spider, each variable v_i is determined using the method of roulette-wheel selection.

Once a new spider s_{new} was formed is compared with the worst spider s_{worst} from the colony according to their weights, where $w_{worst} = min_{l \in \{1,2,\ldots,N\}}(w_l)$. If the new spider s_{new} is better than the worst spider s_{worst}, then s_{worst} is replaced by s_{new}. Otherwise, the new spider is discarded and the colony does not suffer alterations. If a replacement occur, the new spider takes the genre and index from the spider replaced.

4 Optimization Algorithm Based on Social Behaviour Spiders for Clustering Problems

As proposal we present an SSO (Cuevas et al., 2013) approach to solve the clustering problem. This optimization algorithm based on the social behaviour of spiders is a meta-heuristic algorithm of general purpose, so it is necessary to modify many elements of the algorithms such as the representation of the individuals, calculation of the fitness function, etc. Below are presented the elements on which it was necessary to make modifications to the original algorithm proposed in (Cuevas et al., 2013).

4.1 Representations of Spiders (Individuals)

The first consideration to take into account is the representation of each spider. Each spider (male or female) represents a set of k clusters centers, which is a feasible solution to the problem of clustering.

For instance, let $x = \{(10.5;\ 20.4),\ (15.2;\ 25.0)\}$ be a spider that contains $k = 2$ cluster centers that are $\{(10.5;\ 20.4)$ and $(15.2;\ 25.0)\}$, in this particular case each center has dimension $n = 2$.

Each spider of the initial population was generated taking k random points of a given dataset, where k is the number of cluster to be found.

4.2 Distance between Two Spiders

It is necessary to define the distance between two spiders, since a spider is formed by a set of cluster centers (each center formed by several points) and not by a set of points. So we define the distance between two spiders as the sum of the euclidean distances between their centers of clusters.

For instance, let $a = \{(a_{x_1};\ a_{y_1}),\ (a_{x_2};\ a_{y_2})\}$ and $b = \{(bx_1;\ by_1),\ (b_{x_2};\ b_{y_2})\}$ be two spiders that have $k = 2$ clusters centers, with each center having dimension 2. Then the distance between these two spiders will be:

$$d_{a,b} = d((a_{x_1};\ a_{y_1}),\ (b_{x_1};\ b_{y_1})) + d((a_{x_2};\ a_{y_2}),\ (b_{x_2};\ b_{y_2}))$$

where $d((a_{x_1};\ a_{y_1}),\ (b_{x_1};\ b_{y_1}))$ is the Euclidean distance between the centers $(a_{x_1};\ a_{y_1})$ and $(b_{x_1};\ b_{y_1})$.

4.3 Fitness and Weight of a Spider

The fitness of each spider, which is an indicator of how good is the solution that this spider represents, is calculated using the metric $\mathcal{M}$. The aim of the SSO algorithm is to minimize the fitness of the population. Thus, a spider that has the minimum fitness is the best within the population.

The pseudocode for calculating the fitness of a spider is presented in Algorithm 1. The weight of a spider i was re-defined, because fitness and weight have negative correlation, and it is calculated in the following way:

$$w_i = \frac{worst_s - J(s_i)}{worst_s - best_s}$$

$$best_s = min(\ J(s_k)\ ),\ k \in \{1, 2, \ldots, N\}$$

$$worst_s = max(\ J(s_k)\ ),\ k \in \{1, 2, \ldots, N\}$$

where $J(s_i)$ is the fitness value of the spider i that was calculated using Algorithm 1.

4.4 Mating of Spiders

In the mating stage was defined a mating set T which is formed by a dominant male spider and the female spiders that are within its range of mating. From this set T are created new spiders, a new spider represents a set of cluster centers, where each cluster center is inherited from a spider within the set T. In order to define the spider from which the new spider will inherit a cluster center, it is used the roulette-wheel selection.

Algorithm 1: Algorithm for calculating the fitness of a spider

Input: An array of cluster centers C (spider C); a set D of n-dimensional m points; an integer $k > 0$ that represents the number of clusters

Output: Metric $\mathcal{M}$ of spider C

1 Create the set of empty clusters $G = \{G_1, G_2, \ldots, G_k\}$

2 **foreach** *point x of the set D* **do**

3 Assign the point x to the cluster G_i whose center C_i is the nearest to x;

4 **foreach** *cluster G_i* **do**

5 calculate a new center C_i^*;

6 Calculate the metric $\mathcal{M}$ for the set of clusters G as defined in Section 2;

4.5 Substitution of Spiders

In order to decide which spiders will be replaced by the new spiders produced in the mating stage, also is used the roulette wheel selection method, where spiders of the population with less weight (greater fitness) have more probability to be replaced. It is important to note that the weight of a spider have a negative correlation with respect to its fitness value, since we are working with a minimization problem and not with a maximization problem as originally proposed by (Cuevas et al., 2013).

The pseudocode of our proposal is showed in Algorithm 2.

5 Experiments and Results

To compare the algorithms were taken five dataset from UCI (*UCI Machine Learning Repository*) repository: Balance, Cancer-Int, Dermatology, Diabetes, Iris.

The Balance dataset was generated to model psychological experiments, each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.

Algorithm 2: Social Spider Optimization algorithm for the clustering problem

Input: A dataset D of n-dimensional m points; an integer $k > 0$ that represents the number of clusters

Output: Metric $\mathcal{M}$ of the clusters found

foreach *spider C of population P* **do**
    Choose randomly k points from dataset D and create the array C (spider C) of cluster centers;
Calculate fitness of population P;
Calculate weight of population P;
for $i = 2$ *to numberGenerations* **do**
    Cooperative operator for female spiders;
    Cooperative operator for male spiders;
    Mating operator;
    Replacement of spiders in P;
    Calculate fitness of population P;
    Calculate weight of population P;
Return fitness (metric $\mathcal{M}$) of the best solution found;

The Cancer-int dataset is one of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature. This data set includes 201 instances of one class and 85 instances of another class.

In the Dermatology dataset is shown diagnoses of erythemato-squamos diseases.

Diabetes patient records were obtained from two sources: an automatic electronic recording device and paper records. The automatic device had an internal clock to timestamp events, whereas the paper records only provided "logical time" slots (breakfast, lunch, dinner, bedtime).

Finally Iris contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

More features about the datasets are shown in the Table 1.

For the experiments the number of generations was fixed at 100 for the three algorithms (K-means, GA, SSO). For the case of GA and SSO algorithms the number of elements of their respective population was fixed at 100.

The experiments were performed as follows: for each dataset, the three algorithms (K-means, GA, SSO) were executed 50 times . Each execution of an algorithm returns the metric $\mathcal{M}$ of the best solution found. Then, the following measures were calculated: average, median, minimum and maximum value of the results.

Table 1: Properties of datasets

Dataset	Size	Attributes	Classes
Balance	625	4	3
Cancer-Int	699	9	2
Dermatology	366	34	6
Diabetes	768	8	2
Iris	150	4	3

The results of the experiments for each dataset are shown in tables 2, 3, 4, 5, and 6 where the best results are highlighted in bold.

Table 2: Results of the experiments for the dataset Balance

	K-means	Genetic. Alg.	SSO Alg.
Mean	1426,544	1423,860	**1423,851**
Median	1425,804	**1423,851**	**1423,851**
Minimum	**1423,851**	**1423,851**	**1423,851**
Maximum	1442,669	1424,071	**1423,851**

Table 3: Results of the experiments for the dataset Cancer-Int

	K-means	Genetic Alg.	SSO Alg.
Mean	2824,135	2820,319	**2820,302**
Median	2824,136	**2820,302**	**2820,302**
Minimum	2824,136	**2820,302**	**2820,302**
Maximum	2824,136	2821,138	**2820,302**

5.1 Discussion

In the experiments for the dataset Balance, see Table 2, we can see that SSO algorithm has the best results respect to all measures. Furthermore, respect to the median and minimum values the SSO algorithm has the same values as the GA.

From the results for the dataset Cancer-int, shown in Table 3, we can see that SSO algorithm has the best results respect to all measures. Furthermore, respect to the median and minimum values the SSO algorithm has the same results as the GA.

In the case of Dermatology dataset, shown in Table 4, GA algorithm has the best results respect to all measures. Furthermore, respect to the median and minimum values the SSO algorithm has the same results as the GA this result is similar as the two previous experiments.

Table 4: Results of the experiments for the dataset Dermatology

	K-means	Genetic Alg.	SSO Alg.
Mean	1127,390	**1092,353**	1092,355
Median	1121,087	**1092,341**	1092,356
Minimum	1092,644	**1092,341**	**1092,341**
Maximum	1415,274	**1092,373**	**1092,373**

Table 5: Results of the experiments for the dataset Diabetes

	K-means	Genetic Alg.	SSO Alg.
Mean	52072,244	49160,016	**49159,956**
Median	52072,243	49160,214	**49159,939**
Minimum	52072,244	**49157,441**	**49157,441**
Maximum	52072,244	**49161,999**	49165,111

In the Table 5, we can see results of Diabetes dataset, the results shown that SSO algorithm has the best results respect to all measures. And, respect to the minimum values the SSO algorithm has the same results as the GA.

Finally in the results for the Iris dataset, shown in Table 6, SSO algorithm has the best results respect to all measures too. Furthermore, respect to the median and minimum values the SSO algorithm has the same results as the GA.

5.2 Statistical Analysis

An additional statistical analysis was performed for comparing the algorithms, since we are working with stochastic algorithms.

The following methodology was used: first the Kolmogorov-Smirnov test was applied to determine whether results (of 50 executions) of each algorithm have a normal distribution. After determining that the algorithms do not have normal distribution the non parametric Wilcoxon rank sum test was applied to compare the medians of two algorithms. This methodology was discussed and applied in others works (Demšar, 2006), (Durillo et al., 2009), (Muñoz et al., 2011).

The *Wilcoxon rank sum test* is used to test the null hypothesis (H_0) that the samples (of 50 executions) of two algorithms come from distributions with same medians. If the null hypothesis is rejected the alternative hypothesis is assumed (H_A) that the samples come from distributions with different medians.

A significance level of 5% ($p - value$ less or equal than 0.05) was used for the Wilcoxon rank sum test. If the test is successful then the null hypothesis is rejected and the alternative hypothesis is assumed, this result is shown using the *'s+'* symbol. Otherwise $p - value$ is greater than 0.05

Table 6: Results of the experiments for the dataset Iris

	K-means	Genetic Alg.	SSO Alg.
Mean	102.495	97.223	**97.222**
Median	97.326	**97.222**	**97.222**
Minimum	97.326	**97.222**	**97.222**
Maximum	124.182	97.232	**97.222**

and the null hypothesis is assumed, this results is shown using the symbol *'s-'*.

In the table 7 are shown the results of the statistical test of Wilcoxon between the SSO and K-means clustering algorithms. In this table we can see that there is statistical difference between SSO and K-means algorithms for all cases. So we can conclude that the SSO algorithm presents results significantly better than K-means algorithm.

Table 7: Results of the Wilcoxon rank sum test between the SSO algorithm and the K-means algorithm

Dataset	**SSO Alg.** (median)	**K-means** (median)	
Balance	**1423,851**	1425,804	*s+*
Cancer-Int	**2820,302**	2824,136	*s+*
Dermatology	**1092,356**	1121,086	*s+*
Diabetes	**49159,939**	52072,244	*s+*
Iris	**97,222**	97,326	*s+*

In the table 8 is shown the result of the statistical test of Wilcoxon between the SSO and the GA. In this table we can see that for most cases the SSO and GA algorithms have similar results. Only in the case of Iris dataset exists statistical significance, but we can not conclude that an algorithm is better than another since they have the same median, we can only say that the algorithms have different behavior.

Table 8: Results of the Wilcoxon rank sum test between the SSO algorithm and the genetic algorithm

Dataset	**SSO Alg.** (median)	**GA** (median)	
Balance	1423,851	1423,851	*s-*
Cancer-Int	2820,302	2820,302	*s-*
Dermatology	1092,356	**1092,341**	*s-*
Diabetes	**49159,939**	49160,214	*s-*
Iris	97,222	97,222	*s+*

6 Conclusions and Future Work

In this work, a SSO approach for the clustering problem has been proposed. For evaluating its performance experiments were performed

over 5 datasets from the UCI repository (Balance, Cancer-Int, Dermatology, Diabetes, and Iris), also comparisons were performed with two classic approaches for the clustering problem: the k-means algorithm and a genetic algorithm for clustering.

The experiments showed that the SSO algorithm has better results regarding the algorithm k-means, and regarding the genetic algorithm, the SSO algorithm has equally competitive results. All these results were validated statistically using the non-parametric Wilcoxon rank sum test. Thus, the main contribution of this work was to show that the SSO algorithm can produce competitive results when compared with classic algorithms.

As future works, we will include comparisons with newer algorithms of the literature. Also, it is interesting to include in the experiments bigger datasets. Finally, additional experiments will be performed using other metrics such as the Classification Error Percentage used in others works: (De Falco et al., 2007), (Karaboga and Ozturk, 2011) and (Senthilnath et al., 2011).

References

Erik Cuevas, Miguel Cienfuegos, Daniel Zaldvar, and Marco Prez-Cisneros. 2013. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, 40(16):6374 – 6384.

Ivanoe De Falco, Antonio Della Cioppa, and Ernesto Tarantino. 2007. Facing classification problems with particle swarm optimization. *Applied Soft Computing*, 7(3):652–658.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Juan J Durillo, José García-Nieto, Antonio J Nebro, Carlos A Coello Coello, Francisco Luna, and Enrique Alba. 2009. Multi-objective particle swarm optimizers: An experimental comparison. In *Evolutionary Multi-Criterion Optimization*, pages 495–509. Springer.

John H Holland. 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Dervis Karaboga and Celal Ozturk. 2011. A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied Soft Computing*, 11(1):652–657.

S. Lloyd. 1982. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, Mar.

Ujjwal Maulik and Sanghamitra Bandyopadhyay. 2000. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9):1455 – 1465.

Boris Mirkin. 2012. *Clustering: a data recovery approach*. CRC Press.

Daniel M Muñoz, Carlos H Llanos, LDS Coelho, and Mauricio Ayala-Rincón. 2011. Opposition-based shuffled pso with passive congregation applied to fm matching synthesis. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2775–2781. IEEE.

Satyasai Jagannath Nanda and Ganapati Panda. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 16:1–18.

Dan Pelleg and Andrew Moore. 1999. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–281. ACM.

Steven J Phillips. 2002. Acceleration of k-means and related clustering algorithms. In *Algorithm Engineering and Experiments*, pages 166–177. Springer.

J Senthilnath, SN Omkar, and V Mani. 2011. Clustering using firefly algorithm: Performance study. *Swarm and Evolutionary Computation*, 1(3):164–171.

Michael Steinbach, George Karypis, Vipin Kumar, et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston.

Harley Vera-Olivera and José Luis Soncco-Álvarez. 2016. Algoritmo de optimización basado en el comportamiento social de arañas para clustering. Universidad Nacional de San Antonio Abad del Cusco. Undergraduate Thesis for Obtaining the Engineer Degree in Informatics and Systems.

M vijayalakshmi and M Renuka Devi. 2012. A survey of different issue of different clustering algorithms used in large data sets. In *International Journal of Advanced Research in Computer Science and Software Engineering*, pages 305 – 307.

R. Xu and D.C Wunsch. 2009. *Clustering*. Oxford Wiley.