An Approach Based on Tunicate Swarm Algorithm to Solve Partitional Clustering Problem

Murat Aslan

Abstract—**The tunicate swarm algorithm (TSA) is a newly proposed population-based swarm optimizer for solving global optimization problems. TSA uses best solution in the population in order improve the intensification and diversification of the tunicates. Thus, the possibility of finding a better position for search agents has increased. The aim of the clustering algorithms is to distributed the data instances into some groups according to similar and dissimilar features of instances. Therefore, with a proper clustering algorithm the dataset will be separated to some groups and it's expected that the similarities of groups will be minimum. In this work, firstly, an approach based on TSA has proposed for solving partitional clustering problem. Then, the TSA is implemented on ten different clustering problems taken from UCI Machine Learning Repository, and the clustering performance of the TSA is compared with the performances of the three well known clustering algorithms such as fuzzy c-means, k-means and k-medoids. The experimental results and comparisons show that the TSA based approach is highly competitive and robust optimizer for solving the partitional clustering problems.**

Index Terms—**Clustering, fuzzy c-means, k-means, k-medoid, tunicate swarm algorithm.**

I. Introduction

THE PURPOSE of unsupervised learning technique is to find out potential views from data without any class information and data clustering which is an important unsupervised learning technique is a considerable part of data mining [1, 2]. The intention of the clustering algorithms is to split instances into some groups according to their similar and dissimilar features. If one of the clustering technique is implemented on any dataset, the dataset separated to some groups and it is expected that the similarities of these groups will be in a minimum level [3, 4]. The main goal of the using the previous data samples is to obtain a conclusion, estimate a future statement and in diagnosis process [2]. Data clustering methods are used for various research areas such as marketing [5], text mining [6], financial analysis [7], web analysis [8], image segmentation [9], education [10], bioinformatics [11], medical diagnosis [12], wireless sensor networks [13], data science [14], business [15] and so on.

Many techniques proposed in literature such as hierarchical approaches [16], graph based methods [17, 18], partitional (non-hierarchical) techniques [19-28], density based methods [29] and optimization-based clustering approaches [30-32] for solving data clustering problems [2, 33]. Some of optimization-based local search algorithms are Tabu Search (TS) algorithm [34] and Simulated Annealing (SA) algorithm [35]. In recently, optimization-based metaheuristic algorithms are used for many different optimization problems, because of problem free, has a simple structure and easy adaptable to any optimization problems [36]. Some of these metaheuristic algorithms for solving data clustering problems are such as Genetic Algorithm (GA) [37, 38], Teacher Learning Based Optimization (TLBO) [39], Ant Colony Optimization (ACO) [40], Artificial Bee Colony (ABC) [3, 41], Gravitational Search Algorithm (GSA) [42], Particle Swarm Optimization (PSO) [43], Grey Wolf Optimizer (GWO) [4] and Cuckoo Search (CS) [44, 45] algorithms.

In this study, the tunicate swarm algorithm (TSA) which is one of the swarm behavior based optimization algorithm proposed by Kaur et al. [47] to solve global optimization problems is implemented on ten different partitional clustering problems which are taken from UCI Machine Learning Repository [46]. The remainder section of this paper is detailed as follows: the clustering problem is extended in Section II. In Section III, the partitional clustering techniques such as Fuzzy C-Means, K-Medoids and K-Means are explained. In Section IV, each steps of the Tunicate Swarm Algorithm (TSA) are detailed. In Section V, the experimental outputs are given. The conclusions of experiments are given in Section VI.

II. The Clustering Problem

In the phase of clustering process, the data instances are divided into some sub groups according to their similar and dissimilar attributes. Eventually, the main goal of the data clustering is to obtain some homogeneous sets [3, 4, 45]. Partitional clustering methods are distributed the N data instances to k clusters (sets) in accordance with the similarity of the data instances. When data instances are distributed to clusters, some distance metrics [3, 45] such as Manhattan distance [48], Euclidean distance [49] and Minkowski distance [50] are used to find the best centroids. Therefore, the dataset is distributed homogeneously and the efficiency of the clustering is increased. In my study, the sum of squared

Murat ASLAN, is with Department of Computer Engineering, Faculty of Engineering, Şırnak University, Şırnak Turkey, (e-mail: murataslan@sirnak.edu.tr).

https://orcid.org/0000-0002-7459-3035

Manuscript received March 29, 2021; accepted July 16, 2021.
DOI: 10.17694/bajece.904882

Euclidean (SSE) distance which is calculated by present instance and the cluster center of the present instance is used as the objective function of the algorithms. The mathematical formulation of objective function SSE is given in Equation (1) [4, 21].

$$SSE = \sum_{i=1}^{k} \sum_{x\grave{o}C_i} d\left(x, C_i\right) \tag{1}$$

In Equation (1), C_i indicates the set of k centers , x is the data instance which assigned to the C_i cluster and $d(x, C_i)$ shows the SSE distance between center C_i and x instance.

III. PARTITIONAL CLUSTERING TECHNIQUES

In this study, one of the newest optimization-based metaheuristic algorithm which is called TSA is implemented on a dataset taken from UCI Machine Learning Repository for solving partitional clustering problems, and also some classical partitional clustering methods such as Fuzzy C-Means, K- Medoids and K-Means approaches are also implemented on the same dataset. The steps and framework of these approximate methods are detailed in under this section.

A. K-Means Algorithm

K-Means method is one of the important and simple, and besides an effective partitional clustering algorithms. When K-Means algorithm is executed, it is tried to find the best possible centroid (total centroids is k) for each data instances in N. In K-Means approach the data instances are distributed into the centroids according to their maximum similarities and each instance can belong to just one cluster. The fitness value of K-Means is calculated with Equation (1). The value of the each center (C_i) is calculated with Equation (2) [4].

$$C_i = \frac{1}{N_i} \sum_{j=1}^{N} w_{ji} x_j \tag{2}$$

```
Step 1: Create the centers of  k cluster randomly in upper and lower boundaries.
Step 2: Find the nearest cluster center for each data instances.
Step 3: After the each instance is distributed to the nearest cluster, update the center of the each cluster.
Step 4: Until the center of clusters are being stable or reach the maximum number of iteration, return the Step 2.
```

Fig. 1. The pseudo code of K-Means method [23]

Where N_i indicates the number of instances related to ith cluster. w_{ji} indicates a binary value in a set of [0,1] and if w_{ji} is assigned with ‘1’, it is mean that x_j is associated with ith cluster, otherwise x_j is not a member of ith cluster. The steps of k-Means algorithm detailed by Velmurugan [23] and they are shown in Fig 1. The centers of clusters are generated with a random initializing in K-Means algorithm. Hence, the performance of K-Means clustering method increases or decreases according to the position of starting centers.

B. K-Medoids Algorithm

The framework of K-Medoids method is similar to K-means algorithm. However, the selection strategy for cluster centers are different to each other. When a cluster center is determined in K-Means algorithm, any location in upper and lower bound can be a center of cluster. In K-Medoids method the best medoids are selected as cluster centers and a medoid should be an instance from all dataset [4, 51]. In this study, the Partitioning Around Medoids (PAM) [24] is selected as K-Medoids algorithm. The steps of K-Medoids algorithm are given in Fig. 2.

```
Step 1: Create the centers of  K-Medoids randomly from dataset.
Step 2: Find the nearest cluster center for each data instances.
Step 3: For any medoid (m) instance and non-medoid (o) instance which depended with m; swap m and o, and now o is a potential medoid. After then, the fitness values are calculated according to the Equation (1) for new potential medoid set and update position of k medoids.
Step 4: Until the center of clusters are being stable or reach the maximum number of iteration, return the Step 2.
```

Fig. 2. The pseudo code of K-Medoids method [21, 43]

C. Fuzzy C-Means Algorithm

The framework of Fuzzy C-Means method is similar to K-Means algorithm. Fuzzy C-Means approach is developed by Dunn [27] and an extended version of this algorithm is developed by Bezdek [28]. In traditional clustering algorithm like K-Means and K-Medoids methods each data instance only can distribute to the one cluster. However, in fuzzy algorithm, each instance can belong to the one or more than one data clusters according to the membership degree of each cluster. For each instance, the sum of the membership degree for all clusters is equal to 1. The objective of the Fuzzy C-Means algorithm is given in Equation (3) [4, 25, 26].

$$j_m = \sum_{i=1}^{N} \sum_{j=1}^{k} u_{ij}^{m} \left\| x_i - C_j \right\|^2 \tag{3}$$

In here, N indicates the total data instances, k shows the total number of clusters, x_i shows the ith data instance, m shows a value upper than 1, u_{ij} shows the membership degree of the ith instance in jth cluster and C_j shows the center of the jth cluster. When an iteration is carried out u_{ij} and C_j values are updated by the Equation (4) and (5).

$$u_{ij} = \frac{1}{\sum_{n=1}^{k}\left(\frac{\left\|x_i - C_j\right\|^2}{\left\|x_i - C_n\right\|^2}\right)^{2/m-1}} \qquad (4)$$

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} .\ x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \qquad (5)$$

The value of each membership degree is generated randomly in a range of [0,1] and the constraints are given below:

$$0 \le u_{ij} \le 1 \quad and \quad \sum_{i=1}^{k} u_{ij} = 1 \text{ for each } \mathrm{j} = \{1, 2, \dots \mathrm{k}\}$$

The steps of Fuzzy C-Means algorithm are detailed in Fig. 3.

Step 1: Create membership degree matrix (u) randomly.

Step 2: Calculate the Equation (4) and determine the center of clusters according to the matrix u.

Step 3: Recalculate the matrix u with Equation (4).

Step 4: If the stopping criteria is not met, repeat the Step 2 and Step 3.

Fig. 3. The steps of Fuzzy C-Means algorithm [23]

IV. TUNICATE SWARM ALGORITHM (TSA)

The tunicate swarm optimization method is a population-based swarm approach proposed to solve global optimization problems. Tunicates are shining bio-luminescent and generates a faded blue–green light that can be realized from more than many meters away [52]. Each tunicate has a few millimeters length. There is a collective gelatinous tunic in each tunicate, and thanks to this feature all tunicate are connected to each other. Besides, each tunicate separately gets the water from the present sea and creating the jet propulsion by its open end through atrial siphons. Tunicate change the its current position with a propulsion like fluid jet. This propulsion makes to migrate the tunicates vertically in deep ocean. The most spectacular features of tunicate are their jet propulsion and swarm behaviors [47].

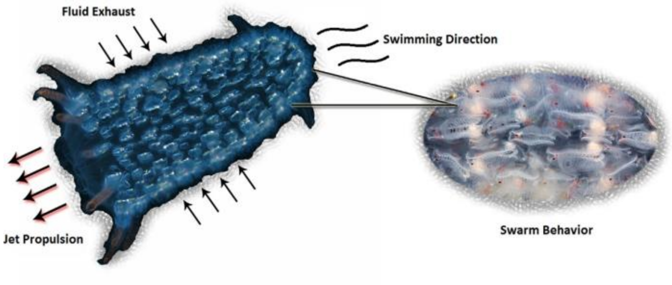

Fig. 4. swarm behavior of tunicate [47]

TSA uses the position of the best tunicate of the population for the purpose of improve the intensification and diversification of the tunicate. Thus, the possibility of finding a better position for search agents has increased. The swarm behavior of TSA is given in Fig. 4. Although tunicate has no idea what is the quality of food source, tunicate has the ability to locate the food source in the sea. tunicate uses Jet propulsion and swarm behavior structures while reaching the food source. Tunicate has three behaviors during Jet propulsion behavior. These behaviors are; Avoiding conflicts between the population of tunicate during the discovering process is that any tunicate constantly try to mobilize itself towards the individual with the best fitness value and try to keep itself close to the best individual. In order to prevent conflicts between tunicates, the vector $\vec{A}$ is used to create the new position of the current tunicate and it is calculated according to the equations given below:

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \qquad (6)$$

$$\vec{G} = c_2 + c_3 - \vec{F} \qquad (7)$$

$$\vec{F} = 2.c_1 \qquad (8)$$

Where, $\vec{G}$ refers to gravity force and $\vec{F}$ indicates to the direction of water in the depths of the ocean. c_1, c_2 and c_3 represent a random reel value generated in the range of [0,1]. The vector $\vec{M}$ is expressed the social forces between the tunicates (search agents) in the search space and calculated according to Equation (9).

$$\vec{M} = \lfloor P_{min} + c_1 \,.\, P_{max} - P_{min} \rfloor \qquad (9)$$

Where, P_{min} and P_{max} values shows the initial and secondary velocities for social interaction, and these values are accepted as 1 and 4, respectively [47]. After conflicts between adjacent tunicates are prevented, search agents begin to move towards adjacent tunicate which has the best fitness value. The movement of present tunicate to the best tunicate is calculated by Equation (10):

$$\vec{PD} = \left|\vec{FS} - r_{rand} \,.\, \vec{P_p}(x)\right| \qquad (10)$$

Where, $\vec{PD}$ refers to the distance between the food source and the search agent, x refers to current iteration, $\vec{FS}$ is the position of the tunicate with the best fitness value, $\vec{P_p}(x)$ is the location of the present individual, and r_{rand} indicates a random reel value generated in the range of [0,1] . The new position of $\vec{P_p}(x')$ is calculated according to Equation (11).

$$\vec{P_p}(x') = \begin{cases} \vec{FS} + \vec{A}.\vec{PD}\ , \text{ if } r_{rand} \geq 0.5 \\ \vec{FS} - \vec{A}.\vec{PD}\ , \text{if } r_{rand} < 0.5 \end{cases} \quad (11)$$

$\vec{P_p}(x')$ shows the new position created for $\vec{P_p}(x)$ according to the position of the best food source $\vec{FS}$. The mathematical model of the swarm behavior for tunicates is explained by Equation (12); the first two best optimum solutions are memorized and the positions of the other search agents are updated according to the location of these recorded best solutions.

$$\vec{P_p}(x+1) = \frac{\vec{P_p}(x)+\vec{P_p}(x+1)}{2+c_1} \quad (12)$$

After position update process, the latest position of tunicate will be in a random location, within a cylindrical or cone-shaped. $\vec{A}$, $\vec{G}$, and $\vec{F}$ promote the location of tunicates to move randomly in a specific search space and prevent the conflicts in tunicates population. The capability exploration and exploitation of TSA is provided by vectors $\vec{A}$, $\vec{G}$, and $\vec{F}$. After these explanations, the steps of TSA are given in Fig. 5.

Step 1: Create the initial tunicate population ($\vec{P_p}$).
Step 2: Determine the control units of TSA and stopping criteria.
Step 3: Compute the fitness values of the initial population.
Step 4: Select the position of the tunicate with the best fitness value.
Step 5: Create the new position for each tunicate by using the Equation (12).
Step 6: Update the position of the tunicates which are out of the search space.
Step 7: Compute the fitness values for the new positions of tunicates.
Step 8: Until stopping criteria is satisfied, repeat steps between 5 to 8.
Step 9: After stopping criteria is satisfied, save the best tunicate position.

Fig. 5. The steps of tunicate swarm algorithm [47]

The flowchart of TSA to solve partitional clustering problem is shown in Fig. 6. It is understood from Fig. 6, firstly, the TSA is initialized with control parameters. Then, the other steps of TSA are executed for solving clustering problem. And finally, the position of the tunicate which has the best fitness value is registered.

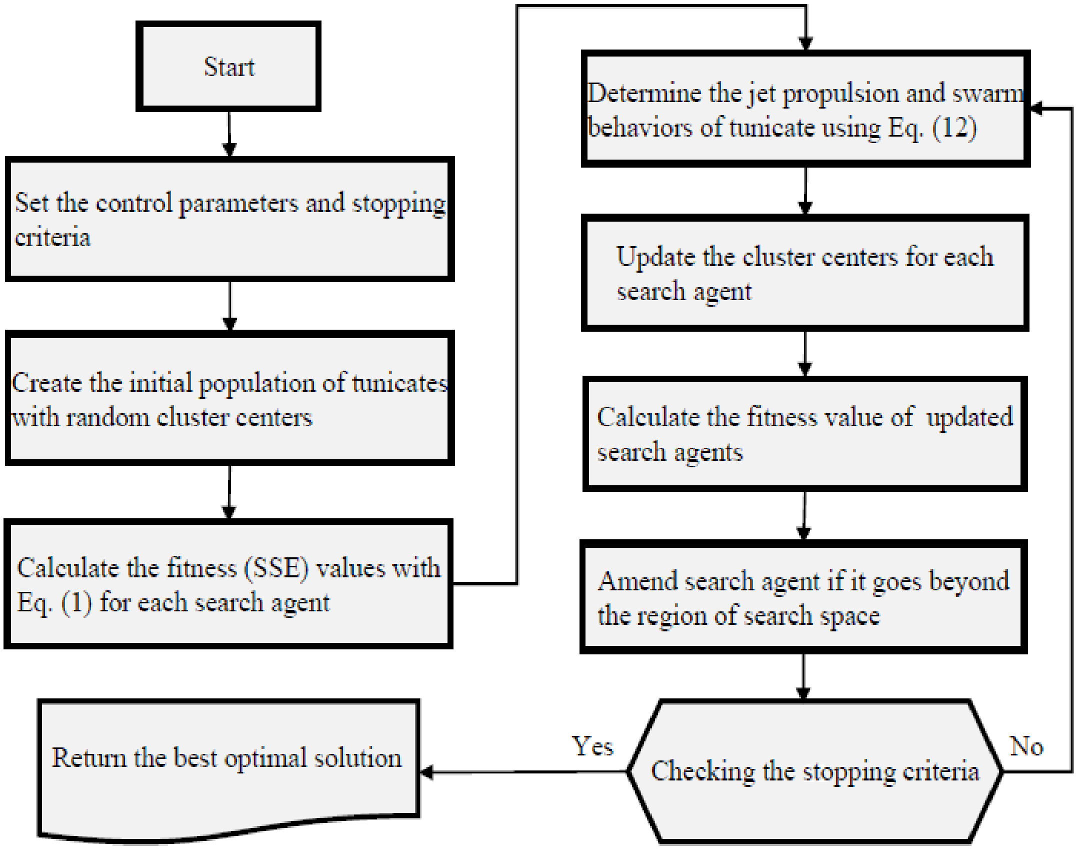
Start
Set the control parameters and stopping criteria
Create the initial population of tunicates with random cluster centers
Calculate the fitness (SSE) values with Eq. (1) for each search agent
Determine the jet propulsion and swarm behaviors of tunicate using Eq. (12)
Update the cluster centers for each search agent
Calculate the fitness value of updated search agents
Amend search agent if it goes beyond the region of search space
Checking the stopping criteria
Yes
No
Return the best optimal solution

Fig. 6. The flowchart of TSA for solving data clustering problem

V. EXPERIMENTAL RESULTS

In this study, TSA is applied on data clustering problem. For experiments, a dataset within 10 instances were taken from UCI Machine Learning Repository [46]. And besides, Fuzzy C-Means (F. C-Means), K-Medoids and K-Means partitional clustering algorithms are implemented on the this dataset. All experiments were carried out on a Windows 10 Pro OS laptop using Intel(R) Core(TM) i7-6700HQ 2.60 GHz CPU, 24 GB of RAM and F.C-Means, K-Medoids, TSA and K-Means algorithms are coded in MATLAB platform. All algorithms are run 30 independent times and the experimental outcomes are reported as best, mean, worst and standard deviation (Std. Dev.) of 30 runs. For all methods the stopping criteria is selected as maximum iteration and it is set to 1000. The population size is set 100 for TSA and the other control parameters of TSA are selected according to the Kaur et al. [47] study.

TABLE I
THE ATTRIBUTES OF THE DATASET [46]

	Data	N. of Clusters	N. of Dimensions	N. of Instances
D1	Balance	3	4	625
D2	Cancer	2	30	569
D3	Cancer-Int	2	9	699
D4	Credit	2	14	690
D5	Dermatology	6	34	366
D6	E. Coli	5	7	327
D7	Glass	6	9	214
D8	Iris	3	4	150
D9	Thyroid	3	5	215
D10	Wine	3	13	178

TABLE II
COMPARISON RESULTS OF TSA WITH K-MEANS, K- MEDOIDS AND F. C-MEANS ALGORITHMS

	Data	Criteria	TSA	K-Means	K-Medoids	F. C-Means
D1	Balance	Best	1424.97	1423.85	1661.82	1722.24
		Worst	1426.39	1433.64	1813.56	1722.24
		Mean	**1425.58**	1425.95	1721.79	1722.24
		Std. Dev.	3.85E-01	2.15E+00	3.94E+01	1.16E-12
		Time	75.406	1.204	0.696	1.275
D2	Cancer	Best	1.34E+154	1.34E+154	1.79E+308	7.63E+156
		Worst	1.34E+154	1.34E+154	1.79E+308	7.63E+156
		Mean	**1.34E+154**	**1.34E+154**	1.79E+308	7.63E+156
		Std. Dev.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		Time	110.937	1.565	0.865	2.339
D3	Cancer-Int	Best	2968.81	2986.96	3311.56	3286.11
		Worst	2973.64	2988.43	4717.47	3286.11
		Mean	**2971.34**	2987.94	3733.51	3286.11
		Std. Dev.	1.24E+00	7.03E-01	3.99E+02	1.08E-12
		Time	86.058	1.412	0.497	1.408
D4	Credit	Best	556749.66	748491.65	558644.51	759180.47
		Worst	556834.46	808744.44	688213.99	759180.47
		Mean	**556769.82**	789553.64	591941.43	759180.47
		Std. Dev.	2.17E+01	2.69E+04	3.49E+04	6.84E-11
		Time	95.497	1.536	0.633	1.748
D5	Dermatology	Best	2247.38	2022.26	2732.90	5196.38
		Worst	2376.46	2197.77	3131.02	5196.38
		Mean	2309.36	**2090.85**	2930.87	5196.38
		Std. Dev.	3.13E+01	5.87E+01	1.04E+02	1.38E-11
		Time	180.005	1.525	4.710	8.680
D6	E. Coli	Best	69.72	66.02	133.02	108.44
		Worst	92.14	70.42	249.46	108.44
		Mean	75.60	**67.52**	160.47	108.44
		Std. Dev.	4.83E+00	1.20E+00	2.13E+01	5.48E-10
		Time	57.812	0.721	1.234	1.574
D7	Glass	Best	303.02	213.42	307.46	400.98
		Worst	372.19	262.57	479.53	404.41
		Mean	341.11	**235.98**	358.90	402.35
		Std. Dev.	1.41E+01	1.48E+01	5.19E+01	1.6042
		Time	54.44	0.515	1.375	1.621
D8	Iris	Best	97.88	97.33	184.54	106.36
		Worst	128.13	122.28	240.30	106.36
		Mean	101.46	**99.00**	210.46	106.36
		Std. Dev.	7.29E+00	6.33E+00	1.62E+01	7.07E-14
		Time	22.340	0.291	0.361	0.310
D9	Thyroid	Best	1886.96	2000.12	2085.73	2812.50
		Worst	2066.64	2024.62	2457.95	2812.50
		Mean	**1916.41**	2011.52	2224.35	2812.50
		Std. Dev.	2.97E+01	8.21E+00	9.43E+01	5.94E-11
		Time	31.546	0.427	0.423	0.492
D10	Wine	Best	16323.45	16555.68	17048.52	17128.46
		Worst	16380.04	18436.95	31007.19	17128.46
		Mean	**16338.28**	18044.71	20981.74	17128.46
		Std. Dev.	1.10E+01	7.60E+02	2.88E+03	6.19E-12
		Time	40.718	0.416	0.494	0.674

The objective function is selected as Equation (1) for all methods. The attributes of the dataset taken from UCI Machine Learning Repository are given in Table I, and in Table I, *N. of Clusters* indicates the total number of clusters, *N. of Dimensions* shows the size of problem and *N. of Instances* shows the size of samples. The obtained experimental outcomes of F.C-Means, K-Medoids, TSA and K-Means algorithms are given in Table II. As seen from Table II, TSA obtains the best mean results for 6 problems (the total number of data instances is 10) such as Balance, Cancer, Cancer-Int, Credit, Thyroid and Wine samples, and TSA finds the second best results for Dermatology, E. Coli, Glass and Iris samples. K-Means algorithm is found the best mean results for 5 problems such as Cancer, Dermatology, E. Coli, Glass and Iris samples. In the light of these obtained experiments, it is shown that the performance of the TSA is higher than F.C-Means, K-Medoids and K-Means algorithms. K-Means algorithm reaches the second best results and the experimental results of F. C-Means algorithm are the worst results among the compared algorithms. In addition, the running time of F.C-Means, K-Medoids, TSA and K-Means algorithms are also given in Table II. It is understood that TSA uses up more time than the other compared algorithms in terms of running time of the methods. The reason of TSA consumes more time is TSA one of the population based metaheuristic algorithm. Metaheuristic algorithms are problem free and they can easily adaptable any problem in generally. But approximate algorithms such as K-Means, K-Medoids and F. C-Means algorithms are proposed for solving clustering problems and they cannot adaptable any other problem in generally. So, it is expected the running time of the approximate methods less than metaheuristic algorithms.

TABLE III
COMPARISON RESULTS OF TSA WITH K-MEANS, K- MEDOIDS AND F. C-MEANS ALGORITHM

Data	Criteria	TSA	K-Means	K-Medoids	F.C-Means
D1	Mean	1425.58	1425.95	1721.79	1722.24
	Rank	**1**	2	3	4
D2	Mean	1.34E+154	1.34E+154	1.79E+308	7.63E+156
	Rank	**1**	**1**	4	3
D3	Mean	2971.34	2987.94	3733.51	3286.11
	Rank	**1**	2	4	3
D4	Mean	556769.82	789553.64	591941.43	759180.47
	Rank	**1**	4	2	3
D5	Mean	2309.36	2090.85	2930.87	5196.38
	Rank	2	**1**	3	4
D6	Mean	75.60	67.52	160.47	108.44
	Rank	2	**1**	4	3
D7	Mean	341.11	235.98	358.90	402.35
	Rank	2	**1**	3	4
D8	Mean	101.46	99.00	210.46	106.36
	Rank	2	**1**	3	4
D9	Mean	1916.41	2011.52	2224.35	2812.50
	Rank	**1**	2	3	4
D10	Mean	16338.28	18044.71	20981.74	17128.46
	Rank	**1**	2	4	3

For a comprehensive analysis of experimental results of compared algorithms, the mean results of 30 run and the rank values of F.C-Means, K-Medoids, TSA and K-Means are shown in Table III. A comparison of rank rate of TSA, K-Means, K-Medoids and F. C-Means algorithms is given in Fig. 7. When the sum of rank rates dedicated in Fig. 7 are considered, the total rank rate of the TSA is 14, and TSA has shown a more powerful yield compared with F.C-Means, K-Medoids and K-Means. And according to the rank results, K-Means algorithm has shown the second best performance and the rank rate of K-Means is 17, the third best solver algorithm is the K-Medoids and the rank rate of K-Medoids is 33, and the performance of F. C Means algorithm is less than the other compared algorithms with 35 rank value.

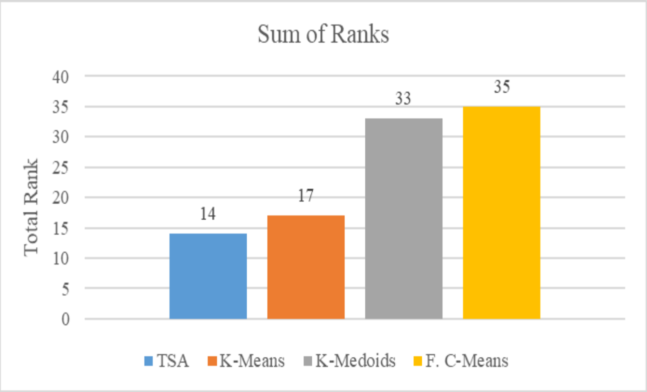

Fig. 7. The total ranks of compared algorithms

VI. CONCLUSIONS

In this study, an approach based on TSA which is newly proposed a population-based swarm optimizer for solving global optimization problems has implemented for solving partitional clustering problem. The experimental outcomes of TSA is compared with three state-of-art classical clustering algorithms: F .C-Means, K-Medoids and K-Means algorithms. In accordance with the analysis and comparisons, the TSA based clustering approach finds a better or comparable performances than K-Means, K-Medoids and F .C-Means algorithms on the UCI Machine Learning Repository dataset in terms of solution quality and robustness.

REFERENCES

[1] A.K. Jain, Data clustering: 50 years beyond k-means, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 3-4.

[2] A. Kaur, Y. Kumar, A new metaheuristic algorithm based on water wave optimization for data clustering, Evolutionary Intelligence, (2021) 1-25.

[3] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Applied soft computing, 11 (2011) 652-657.

[4] M. Karakoyun, O. İnan, İ. Akto, Grey Wolf Optimizer (GWO) Algorithm to Solve the Partitional Clustering Problem, International Journal of Intelligent Systems and Applications in Engineering, 7 (2019) 201-206.

[5] V. Holý, O. Sokol, M. Černý, Clustering retail products based on customer behaviour, Applied Soft Computing, 60 (2017) 752-762.

[6] L.M. Abualigah, A.T. Khader, M.A. Al-Betar, O.A. Alomari, Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering, Expert Systems with Applications, 84 (2017) 24-36.

[7] Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, Expert Systems with Applications, 36 (2009) 10604-10611.

[8] S. Gong, W. Hu, H. Li, Y. Qu, Property Clustering in Linked Data: An Empirical Study and Its Application to Entity Browsing, International Journal on Semantic Web and Information Systems (IJSWIS), 14 (2018) 31-70.

[9] A. Mekhmoukh, K. Mokrani, Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation, Computer methods and programs in biomedicine, 122 (2015) 266-281.

[10] Á.A.M. Navarro, P.M. Ger, Comparison of clustering algorithms for learning analytics with educational datasets, IJIMAI, 5 (2018) 9-16.

[11] I. Triguero, S. del Río, V. López, J. Bacardit, J.M. Benítez, F. Herrera, ROSEFW-RF: the winner algorithm for the ECBDL'14 big data competition: an extremely imbalanced big data bioinformatics problem, Knowledge-Based Systems, 87 (2015) 69-79.

[12] L. Wang, X. Zhou, Y. Xing, M. Yang, C. Zhang, Clustering ECG heartbeat using improved semi-supervised affinity propagation, IET Software, 11 (2017) 207-213.

[13] J. Zhu, C.-H. Lung, V. Srivastava, A hybrid clustering technique using quantitative and qualitative data for wireless sensor networks, Ad Hoc Networks, 25 (2015) 38-53.

[14] R. Hyde, P. Angelov, A.R. MacKenzie, Fully online clustering of evolving data streams into arbitrarily shaped clusters, Information Sciences, 382 (2017) 96-114.

[15] C.-H. Chou, S.-C. Hsieh, C.-J. Qiu, Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction, Applied Soft Computing, 56 (2017) 298-316.

[16] J. Han, M. Kamber, J. Pei, Data mining concepts and techniques third edition, The Morgan Kaufmann Series in Data Management Systems, 5 (2011) 83-124.

[17] S. Schaeffer, Graph clustering. Comput. Sci. Rev. 1 (1), 27–64, in, 2007.

[18] B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica chimica acta, 1097 (2020) 37-48.

[19] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert systems with applications, 40 (2013) 200-210.

[20] J.A. Hartigan, M.A. Wong, AK-means clustering algorithm, Journal of the Royal Statistical Society: Series C (Applied Statistics), 28 (1979) 100-108.

[21] P. Arora, S. Varshney, Analysis of k-means and k-medoids algorithm for big data, Procedia Computer Science, 78 (2016) 507-512.

[22] M. Capó, A. Pérez, J.A. Lozano, An efficient approximation to the K-means clustering for massive data, Knowledge-Based Systems, 117 (2017) 56-69.

[23] T. Velmurugan, Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data, Applied Soft Computing, 19 (2014) 134-146.

[24] L. Kaufman, P.J. Rousseeuw, Partitioning around medoids (program pam), Finding groups in data: an introduction to cluster analysis, 344 (1990) 68-125.

[25] J. Jędrzejowicz, P. Jędrzejowicz, Distance-based online classifiers, Expert Systems with Applications, 60 (2016) 249-257.

[26] X. Qiu, Y. Qiu, G. Feng, P. Li, A sparse fuzzy c-means algorithm based on sparse clustering framework, Neurocomputing, 157 (2015) 290-295.

[27] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, (1973).

[28] J.C. Bezdek, Objective function clustering, in: Pattern recognition with fuzzy objective function algorithms, Springer, 1981, pp. 43-93.

[29] A. Moreira, M.Y. Santos, S. Carneiro, Density-based clustering algorithms–DBSCAN and SNN, University of Minho-Portugal, (2005) 1-18.

[30] S.J. Nanda, G. Panda, A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm and Evolutionary computation, 16 (2014) 1-18.

[31] A. Nayyar, N.G. Nguyen, Introduction to swarm intelligence, Advances in swarm intelligence for optimizing problems in computer science, (2018) 53-78.

[32] A. Nayyar, S. Garg, D. Gupta, A. Khanna, Evolutionary computation: theory and algorithms, Advances in swarm intelligence for optimizing problems in computer science, (2018) 1-26.

[33] S. Saraswathi, M.I. Sheela, A comparative study of various clustering algorithms in data mining, International Journal of Computer Science and Mobile Computing, 11 (2014) 422-428.

[34] C.S. Sung, H.W. Jin, A tabu-search-based heuristic for clustering, Pattern Recognition, 33 (2000) 849-858.

[35] S.Z. Selim, K. Alsultan, A simulated annealing algorithm for the clustering problem, Pattern recognition, 24 (1991) 1003-1008.

[36] M. Aslan, M. Gunduz, M.S. Kiran, JayaX: Jaya algorithm with xor operator for binary optimization, Applied Soft Computing, 82 (2019) 105576.

[37] M.A. Rahman, M.Z. Islam, A hybrid clustering technique combining a novel genetic algorithm with K-Means, Knowledge-Based Systems, 71 (2014) 345-365.

[38] Y. Marinakis, M. Marinaki, M. Doumpos, N. Matsatsinis, C. Zopounidis, A hybrid stochastic genetic–GRASP algorithm for clustering analysis, Operational Research, 8 (2008) 33-46.

[39] Y. Kumar, P.K. Singh, A chaotic teaching learning based optimization algorithm for clustering problems, Applied Intelligence, 49 (2019) 1036-1062.

[40] P. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta, 509 (2004) 187-195.

[41] G. Sahoo, A two-step artificial bee colony algorithm for clustering, Neural Computing and Applications, 28 (2017) 537-551.

[42] X. Han, L. Quan, X. Xiong, M. Almeter, J. Xiang, Y. Lan, A novel data clustering algorithm based on modified gravitational search algorithm, Engineering Applications of Artificial Intelligence, 61 (2017) 1-7.

[43] A. Khatami, S. Mirghasemi, A. Khosravi, C.P. Lim, S. Nahavandi, A new PSO-based approach to fire flame detection using K-Medoids clustering, Expert Systems with Applications, 68 (2017) 69-80.

[44] A. Bouyer, A. Hatamlou, An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms, Applied Soft Computing, 67 (2018) 172-182.

[45] S.I. Boushaki, N. Kamel, O. Bendjeghaba, A new quantum chaotic cuckoo search algorithm for data clustering, Expert Systems with Applications, 96 (2018) 358-372.

[46] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.html, in, 2021.

[47] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, Engineering Applications of Artificial Intelligence, 90 (2020) 103541.

[48] S.N. Neyman, B. Sitohang, S. Sutisna, Reversible fragile watermarking based on difference expansion using manhattan distances for 2d vector map, Procedia Technology, 11 (2013) 614-620.

[49] D.P. Mesquita, J.P. Gomes, A.H.S. Junior, J.S. Nobre, Euclidean distance estimation in incomplete datasets, Neurocomputing, 248 (2017) 11-18.

[50] M. Luo, B. Liu, Robustness of interval-valued fuzzy inference triple I algorithms based on normalized Minkowski distance, Journal of Logical and Algebraic Methods in Programming, 86 (2017) 298-307.

[51] H.-S. Park, C.-H. Jun, A simple and fast algorithm for K-medoids clustering, Expert systems with applications, 36 (2009) 3336-3341.

[52] J. Berrill, The Tuniccafa, The Royal Society: London, (1950).

BIOGRAPHIES

Murat Aslan received B.Sc. and M.Sc. degrees in Computer Engineering from Selçuk University in 2011 and 2017 respectively and received Ph.D. degrees in Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Konya Technical University in 2020. Recently, he is an Assistant Professor in the Department of the Computer Engineering at Şırnak University and his current interests include applications of Graph Theory, Discrete and Binary Optimization Problems, Swarm Intelligence or Evolutionary Computation Algorithms.