Multi-objective PSO with Pareto Neighborhood topology for Clustering

Yassmine Soussi ^[1,2] , (*) Nizar Rokbani^[2,3], Seyedali Mirjalili ^[4,5] , Ali Wali^[2], Adel.M Alimi^[2]

¹ University of Sousse, ISITCom, 4011, Sousse, Tunisia

² University of Sfax, National Engineering School of Sfax (ENIS), REGIM-Lab: REsearch Groups in Intelligent Machines, LR11ES48, 3038, Sfax, Tunisia

³ University of Sousse, High Institute of Applied Science and technology of Sousse, Tunisia

⁴ Centre for Artificial Intelligence Research and Optimization at Torrens University, Australia

⁵Yonsei Frontier Lab, Yonsei University, Seoul, Korea

ABSTRACT

In this paper a new technique is integrated to Multi-Objective Particle Swarm Optimization (MOPSO) algorithm, named Pareto Neighborhood (PN) topology, to produce MOPSO-PN algorithm. This technique involves iteratively selecting a set of best solutions from the Pareto-Optimal-Fronts and trying to explore them in order to find better clustering results in the next iteration. MOPSO-PN was then used as a Multi-Objective Clustering Optimization (MOCO) Algorithm, it was tested on various datasets (real-life and artificial datasets). Two scenarios have been used to test the performances of MOPSO-PN for clustering: In the first scenario MOPSO-PN utilizes, as objective functions, two clusters validity index (Silhouette-Index and overall-cluster-deviation), three datasets for test, four algorithms for comparison and the average Minkowski Score as metric for evaluating the final clustering result; In the second scenario MOPSO-PN used, as objectives functions, three clusters validity index (I-index, Con-index and Sym-index), 20 datasets for test, ten algorithms for comparison and the F-Measure as metric for evaluating the final clustering result. In both scenarios, MOPSO-PN provided a competitive clustering results and a correct number of clusters for all datasets.

Keywords— *Clustering, Multi-Objective Clustering Optimization (MOCO), Multi-Objective Particle Swarm Optimization Algorithm (MOPSO), Pareto Neighborhood (PN), Multi-objective PSO with Pareto Neighborhood topology (MOPSO-PN).*

1. INTRODUCTION

Clustering, or unsupervised classification, is a major research topic in machine learning, data analysis, data mining and pattern recognition. It is an integral part of a whole process of exploratory data analysis making it possible to produce tools for synthesizing, predicting, visualizing and interpreting a data set to build homogeneous groups (clusters) of individuals, based on their similarities or dissimilarities [17] [18], in the sense that two close individuals must belong to the same group, while two distant individuals must belong to different groups.

The problem of clustering tends to be more difficult than classification since no information is provided on the number of clusters “k”, their centers, and configurations. Many multi-objective optimization algorithms have been used in the clustering domain, including Genetic Algorithms [27]. Simple Genetic Algorithm SGA is the first mono-objective version of Genetic Algorithms [10]. Multi-objective Genetic Algorithm, MOGA [11], is the multi-objective version of genetic algorithms.

Before continuing with the description of some multi-objective clustering methods, we should note that the majority of those methods deals with the compactness, the total symmetry and the connectedness of clusters. The compactness of the clustering measures how close are the objects that belong to the same cluster, the total symmetry of the clusters evaluates how the clusters candidates are symmetry distributed regarding the center while the connectedness of the clusters measures how separated clusters are connected

Multi-objective Particle Swarm Optimization and Simulated Annealing algorithm, MOPSOSA was used in [1]. To product a goods clustering solutions, this algorithm optimizes simultaneously three cluster validity indices: the DB-Index which takes into account the compactness of the clustering and is established using the Euclidean distance, the Sym- Index which is looking for the total symmetry and is centered on the point symmetry distance and finally the Con-Index which examines the connectedness and is established using the short distance. The MOPSOSA was tested on UCI benchmarks datasets including 5 real-life datasets (Iris, Cancer, Newthyroid, LiverDisorder and Glass) and artificial datasets including 14 datasets (Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty and Fourty).

GenClustMOO and GenClustPESA2 are two multi-objective clustering algorithms proposed in [31]; these algorithms use three cluster validity indices to optimize cluster-compactness, cluster-connectedness and total-symmetry. To evaluate the performances of GenClustMOO and GenClustPESA2, 19 artificial datasets (Sym-5-2, Sym-3-2, Ellip-2-2, Ring-3-2, Rect-3-2, Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty and Fourty) and 7 real-life datasets (Iris, Cancer, Newthyroid, Wine, LiverDisorder, LungCancer and Glass) are used.

Multi-Objective Clustering with automatic number of clusters (k) determination, MOCK [22], applied for web mining, this algorithm optimizes two objective functions the Overall Cluster Deviation and the Connectivity, which are based respectively on the compactness and connectedness of the cluster. To evaluate MOCK, 9 test datasets are employed (Square1, Square4, Sizes5, Triangle1, Long1, Twenty, D2C10, Spiral and LongSquare).

Variable string length Genetic Algorithm with Point Symmetry-Based clustering, VGAPS-clustering [5], based on symmetry-index (sym-index). This algorithm is tested on 5 real-life datasets (Iris, Cancer, Newthyroid, Liver Disorder and Glass) and 14 artificial datasets (Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty and Fourty).

A multi-objective artificial bee optimization algorithm called cOptBees-MO was proposed in [8], in which two scenarios were used. The first scenario tries to optimize two cluster validity indices

(Silhouette and Overall Cluster Deviation), while the second attempts to optimize three cluster validity indices: the I-Index, the Con-Index and the Sym-Index. The evaluation of cOptBees-MO is based on 10 real-life datasets: three of them are used in the first scenario (Votes, Zoo and Soybean) and the rest (Iris, Cancer, Newthyroid, Wine, LiverDisorder, LungCancer and Glass) are used in the second scenario. A Hybrid Chain-Hypergraph P System for Multi-objective Ensemble Clustering, HCHPS-MOEC, was proposed in [37]. This system combines the advantages of chain and hypergraph topology to establish three types of sub-systems: the reaction-chain-membrane subsystems, the local-communication-membrane subsystems and global-ensemble-membrane subsystems. The reaction-chain-membrane subsystem is used to implement the multiobjective clustering strategies using three new types of rules: the Nondominated-Object-Selection-Rules, the Crossover-Rules and the Mutation-Rules). The evaluation of HCHPS-MOEC is based on two artificial datasets and 17 UCI-real-life datasets (Iris, Newthyroid, Wine, Diabetes, Bupa, Yeast, Glass, Cancer, Heartstatlog, Balancescale, Seeds, Aggregation, Vowel, WBC, Ecoli, Zoo and Heart).

A Reference Vector-based Multi-Objective Clustering for high-dimensional data, RVMOC, was proposed in [21] to optimize three multiple clustering criteria (the intra cluster dispersion, the inter cluster separation and the negative shannon entropy) using the pareto fronts with a fuzzy estimation of the best returned solution around the knee point with evaluation over the UCI benchmarks datasets [20] including 15 real-life datasets (Australian, Auto MPG, Balance, BCWO, Dermatology, Ecoli, Glass, Heart, Image, Iris, Mice protein, Pen, Thyroid, Wholesale and Wine) and gene-expression datasets [25] including 6 datasets (CNS tumors, Leukemia, Lung cancer, Normal, Novartis and St. Jude).

A Multi-objective Cat Swarm Optimization Clustering Analysis for Data Partition, MOCSO, was proposed in [35]. This algorithm uses the cat swarm for searching the optimal clustering solution which is presented by the position of a cat. During the search phase, the cat updates its position through two modes: the seeking-mode and the tracing-mode, using respectively the simulated annealing strategy and the quantum theory. As objective functions, MOCSO use the cohesion and the connectivity. For Evaluation, MOCSO uses three artificial datasets (DS1 [14], DS2 [34] and DS3 [14]), four UCI-real-life datasets (Iris, WDBC, Wine and Bcw) and a field working condition dataset.

A Multi-Objective Quantum Moth Flame Optimization MOQMFO algorithm which was proposed by [33]. MOQMFO combines the features of the Quantum theory and the multi-objective Moth Flame Optimization (MOMFO) which is based on the flying mood of Moths while looking for lights sources (Flames). MOQMFO was developed through two scenarios as in

cOptBees-MO in order to provide a good clustering results and the real number of clusters of each dataset. The evaluation of MOQMFO is based on 6 real-life datasets (Iris, Cancer, Newthyroid, LiverDisorder, Glass and Wine) and 14 artificial datasets (Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty and Fourty). Here The evaluation of MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, VGAPS, cOptBee-MO and MOQMFO is based on F-Measure, Average F-measure and number of clusters produced by each of them; The evaluation of MOCSO is based only on F-measure; While the evaluation of HCHPS-MOEC, RVMOEC is fully metric, based only on average F-measure, which mean not help in comparing them to others methods of clustering. Looking to the results provided by those predefined algorithms, we can see that: MOPSOSA, GenClustMOO, MOQMFO, are able to detect the correct number of clusters for all dataset while GenClustPESA2, MOCK, VGAPS, cOptBee-MO did it in a wrong way for some datasets. Concerning F-measure values and average F-measure values we can see that: MOPSOSA, GenClustMOO and MOQMFO provided a good values of F-measure and average F-measure; GenClustPESA2, MOCK and VGAPS, cOptBee-MO and MOCSO provided a mid-range values of those latter.

The gaps of non-detection of the exact number of clusters and good values of F-measure, met with some of those predefined methods, motivate a proposition of a new multi-objective method with a special focus on clustering and an attempt to overcome all those gaps. This paper proposes a multi-objective clustering algorithm using Particle Swarm Optimization (PSO) [32] with a new topology of neighborhood exploration based on Pareto-Optimal-Solutions called Multi-objective Clustering using Particle Swarm Optimization based Pareto Neighborhood topology MOPSO-PN. The Pareto Neighborhood topology allows selecting a set of best solutions from the Pareto-Optimal-Fronts and exploring them in order to find better clustering results in the next iteration. The MOPSO-PN used clusters validities indices [8] as objectives functions in the process of multi-objective clustering in order to help to find an optimal number of clusters and an optimal clustering solution and F-measure [31] as metric for evaluation of the final obtained clustering solutions. Based on this metric, the performance of MOPSO-PN is then compared to the predefined state of art algorithms. This paper is organized as follows: In the second section a representation of clustering as well as multi-objective clustering (its bases and objective functions) is made; In section 3, the proposed method MOPSO-PN for clustering is detailed; In section 4, experimental investigations of MOPSO-PN on different databases are discussed, and compared to ten similar proposals; and finally in section 5, conclusion and perspectives are elaborated.

2. MULTI-OBJECTIVE CLUSTERING OPTIMIZATION (MOCO)

Clustering consists in decomposing a dataset (D) into several non-empty homogeneous subsets called clusters $\{C_1, C_2, \dots, C_k\}$, where k is the number of clusters which may or may not be known a priori (Jain, 2010). Data in each cluster are sharing common features according to similarity criteria (Jain & Dubes, 1988).

A data-element $d \in D$ is found in one and only one cluster C_i .

Let P be a family of sets ($P = \{C_1, C_2, \dots, C_k\}$): P is a partition of D if only if:

- The empty set is not in P : $\emptyset \notin P$;
- The union of the sets of P is D ; $D = C_1 \cup C_2 \cup \dots \cup C_k$;
- The intersection of any two sets of P is the empty set $\emptyset$: $C_i \cap C_j = \emptyset$, $i \neq j$, $i = 1, 2, \dots, k$, $j = 1, 2, \dots, k$.

2.1. Definitions

Multi-objective optimization problem [7] [6] [9] [16] aims to optimize several contradictory objectives functions $\{F_1, F_2, \dots\}$. Given a problem with a couple of objectives functions F_1 and F_2 subject to minimization; The multi-objective optimization of this problem may be formulated by the following:

- Find x belonging to [Lower-Bound; Upper-Bound]:
- Satisfying: $\min (F_1(x)); \min (F_2(x)) \dots \min (F_n(x))$

In MOCO problems, cluster validity indices are generally used as objectives functions. The key concepts of MOCO are detailed in the next paragraphs.

2.1.1. Domination:

In Multi-Objective Optimization (MOO) [12], the domination concept is used to compare solutions of the problem [16]. Given two clustering solutions S_1 and S_2 in minimization case; S_1 is said to be better than S_2 if only if S_1 dominates S_2 and that's performed if only if all objectives functions of S_1 have better value than those of S_2 , it's mean that all objective functions have lower values, or if all objectives functions of S_1 are equal to those of S_2 but only one objective functions of S_1 is lower than that of S_2 . For Example in minimization case with x objective functions.

S_1 is said to dominate S_2 , denoted as $S_1 < S_2$, iff:

$$1. \forall i \in \{1, \dots, x\} : S_1(i) \leq S_2(i) \quad (1)$$

$$2. \exists j \in \{1, \dots, x\} : S_1(j) < S_2(j) \quad (2)$$

2.1.2. Pareto Optimal Solutions

In MOO problems (for example in minimization problem with x objective functions) we cannot declare a point as a solution to the problem because if we find a point with the best minimum value for an objective function there is certainly another objective function relating to this point whose value is not minimal. Consequently, the solution to MOO problem may not be a single solution but rather a subset of solutions, from the set of solutions in the design space, represented by a curve in which each solution is optimized according to a single objective function among the functions relating to the problem. The set of solutions of this curve are called Pareto-Optimal-Solutions, they are the non-dominated solutions in the sense that there is no other solution in the design space better than them or can dominate them if all the objective functions are considered. Therefore solutions which are worse than the Pareto-Optimal-Solutions are called the dominated solutions. [16].

2.1.3. Repository

The repository is a data structure containing the Pareto-Optimal-Solutions which are non-dominated solutions of the problem as in [6] [16]. It is iteratively updated with new optimal solutions generated at each iteration. The update is conditioned by new repository members shouldn't being dominated by the old members.

2.2. Objective functions in MOCO

Objectives functions in MOCO are presented by cluster validity indices. In this paper, MOPSO-PN algorithm used, as objective functions, five cluster validity indices: Silhouette, Overall Cluster Deviation ($Dev(C)$), I-Index, Con-Index; and Sym-Index.

Variables used in the descriptions of these cluster validity indices are presented in table1.

Table 1. Description of variables used in the equations of cluster validity indices

Variable	signification
n	The number of object in the dataset
d	The dimension of the dataset (number of features)
K	The real number of clusters in the dataset
$\vec{x}_i$	The i-th object in the dataset
n_k	The number of objects in Cluster (k)
$\vec{C}_k$	The centroid of Cluster (k)
$d_e(,)$	The euclidean distance
$\vec{x}_i^k$	The i-th object of the k-th cluster

2.2.1. Silhouette

The silhouette method tends to measure how an object is belonging to a given cluster; its maximal value is (+1) meaning that the object is confused with its cluster while a minimum of (-1) indicates the opposite [8] [24].

The silhouette index is calculated using equation 3.

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (3)$$

a_i is the mean distance of an object $\vec{x}_i$ to other objects belonging to the same cluster of $\vec{x}_i$;

b_i is the minimum of the mean distances of $\vec{x}_i$ to other objects belonging to the same cluster of $\vec{x}_i$.

As illustrated in equation 4, for any cluster its silhouette index is the mean of the silhouette indices of all objects. [15]

$$S = \frac{1}{n} \sum_{i=1}^n S_i \quad (4)$$

2.2.2. Overall cluster deviation (Dev(C))

The cluster deviation (Dec(C)) as presented in [8] [14] and is the metric evaluating the compactness of a cluster. Dec(C) is the overall summed Euclidean distances between cluster centers and their corresponding data objects and is calculated as follows:

$$Dev(C) = \sum_{k=1}^K \sum_{i=1}^{n_k} d_e(\vec{C}_k, \vec{x}_i) \quad (5)$$

where C is the set of clusters, K, n_k and $\vec{C}_k$ present respectively the number of clusters, the number of objects in Cluster (k) and the centroid of Cluster (k); $d_e(, ,)$ presents the Euclidean Distance between the centroid $\vec{C}_k$ and his corresponding object $\vec{x}_i$.

Dev(C) should be minimized since a lower value gives optimal clustering.

2.2.3. I-index

I-index measures the geometric repartition of clusters related to their centers using the Euclidean distance, [8] [31]:

$$I(K) = \left(\frac{1}{K} \times \frac{\varepsilon_1}{\varepsilon_k} \times D_k \right)^p \quad (6)$$

where ε_1 , is constant for a given data, the power p is used to control the contrast between the different cluster configurations,

$$\varepsilon_k = \sum_{k=1}^K \sum_{j=1}^{n_k} d_e(\vec{C}_k, \vec{x}_j^k) \quad (7)$$

$$D_k = \max_{i,j=1}^K d_e(\vec{C}_i, \vec{C}_j) \quad (8)$$

In equations (7) and (8) $\vec{C}_k$ is the center of the Cluster (k), $\vec{x}_j^k$ stands for object (j) of Cluster (k).

In this paper we used $E_1=10$ and $p = 2$ as in [23] and [31].

2.2.4. Con-index

This cluster validity index informs about the connectivity, it is based on the relative neighborhood, which is a graph measuring how separated clusters are connected [8] [30], See equation (9).

$$Con = \frac{\sum_{k=1}^K \sum_{j=1}^{n_k} d_{short}(\vec{C}_k, \vec{x}_j^k)}{n \times \min_{k,j=1 \wedge k \neq j}^K d_{short}(\vec{C}_k, \vec{C}_j)} \quad (9)$$

Here, $d_{short}(\vec{C}_k, \vec{x}_j^k)$ defines the minimum distance in the neighborhood graph between the center $\vec{C}_k$ of cluster (k) and the object $\vec{x}_j^k$ ($\vec{x}_j^k$ is the j -th object of cluster (k)). A minimal value of Con-index means that the cluster is compact and separated from its neighbors.

2.2.5. Sym-index

The Symmetry index, Sym-index, evaluates the symmetry of a cluster in regards to its center. Sym-index as defined in [8] [4] is formulated by equation (10):

$$Sym(K) = \left(\frac{1}{K} \times \frac{1}{E_k} \times D_k \right) \quad (10)$$

Where the internal expressions are detailed in equations (11, 12, 13, 14).

$$E_k = \sum_{i=1}^K E_i \quad (11)$$

Such that

$$E_i = \sum_{j=1}^{n_i} d_{ps}^*(\vec{x}_j^i, \vec{C}_i) \quad (12)$$

$$d_{ps}(\vec{x}_j^i, \vec{C}_i) = d_{sym}(\vec{x}_j^i, \vec{C}_i) \times d_e(\vec{x}_j^i, \vec{C}_i) \quad (13)$$

$$D_k = \max_{i,j=1}^K \|\vec{c}_i - \vec{c}_j\| \quad (14)$$

Where K represents the number of clusters; D_k is the maximum Euclidean distance between two center of clusters (the clusters centers c_i and c_j are taken paired); $d_{\text{sym}}(\vec{x}, \vec{c})$ is a symmetry measure of $\vec{x}$ with respect to $\vec{c}$ and $d_e(\vec{x}, \vec{c})$ is the Euclidean distance between $\vec{x}$ and $\vec{c}$. Sym-index is subject of maximization to get the correct number of clusters.

3. PROPOSED METHODS: MOPSO-PN ALGORITHM

In this section, the MOPSO-PN algorithm is first presented as a heuristic improvement secondly as a multi-objective clustering application.

3.1. PSO Algorithm

PSO [32] is a well-regarded optimization algorithm inspired by the social behavior of animals (particles). In PSO, positions of particle Pos_i present solutions to the problem. The optimization in this algorithm is made by updating the position of each particle Pos_i in the swarm based on velocity Vel_i using equation 16; velocities are also updated using equation 15.

$$\text{Vel}_i(\text{iter}+1) = W \cdot \text{Vel}_i(\text{iter}) + c1 \cdot r1 (\text{pbest}_i - \text{Pos}_i) + c2 \cdot r2 (\text{gbest} - \text{Pos}_i) \quad (15)$$

$$\text{Pos}_i(\text{iter}+1) = \text{Pos}_i(\text{iter}) + \text{Vel}_i(\text{iter}) \quad (16)$$

where Pos_i and Vel_i present respectively the position and the velocity of i -th particle; t presents the number of iteration; $c1$ is the cognitive factor, $c2$ is the social factor; $r1$, $r2$ two numbers chosen randomly from $[0, 1]$, pbest_i is the local best according to a given topology, gbest is the global best position in the swarm, W stands for the inertia weight.

3.2. Multi-Objective PSO

The multi-objective form, MOPSO [7] [16], consists of combining the characteristics of the original algorithm and the multi-objective optimization form, by the application of the concept of “Dominance” to store the best solutions obtained during each generation into a “Repository”: the Repository is a set of non-dominated solutions standing for “Pareto-Optimal-Solutions”. In [9], a non-dominated sorting policy is used to detect the pareto fronts candidates then a crowding distance sorting is used to reduce the repository size for a better detection of the best solution. The selection of the best solution from the “Pareto-Optimal-Solutions” is made through a technique based on the distance between the set of points of the Pareto and an ideal point called Utopia-Point; the best solution is then defined as the closest point to the Utopia-Point [8]. As showed in Figure 1, green circles represent the Pareto-Optimal-Solutions (Repository members), yellow circle presents the utopia point, which is the intersection between the line parallel to $F2$ and which passes through the solution-point with the minimum value of $F1$ and the line parallel to $F1$ and which passes through the solution-point with the minimum value of $F2$, the best

solution which is defined as the closest point to the Utopia-Point is presented by the orange circle.

Figure 1. The selection of best pareto solution using a technique based on the minimal distance from the utopia-point

3.3. The Pareto Neighborhood Topology

Assuming a swarm of $N1$ particles and a repository, including best solutions, of size $N2 < N1$. In classical MOPSO, in each iteration, local solutions in the swarm are handled with a native PSO mechanism, the repository is updated with new optimal solutions from the swarm and the best solution in the repository is assumed to be the closest solution to the utopia point.

The proposed Pareto Neighborhood (PN) intrinsically changes the local solutions of particles in the swarm by changing a given number $N3$ of particle solutions, with solutions from the repository.

More clearly, the PN consists in selecting a neighborhood of $N3$ solutions ($N3 \leq N2$), from the Pareto-Optimal-Solutions (Repository members), closest to the Utopia point and different from the best Pareto solution and assign them to $N3$ solutions of particles in the swarm.

Figure 2 shows how the PN is managed. $N3=5$ this means five best Pareto neighborhood solutions (green circles) closest to the Utopia point (yellow circle) and different from the best Pareto solution (orange circle) are selected from $N2=11$ Pareto-Optimal-Solutions (green circle) and affected to $N3=5$ particles in the swarm of size ($N1=N2*2=22$).

Figure 2. Representation of Pareto Neighborhood topology with $N3=5$ Neighbors.

The PN may be used with any Multi-objective meta-heuristic it allows to limit the side effects of a non-precise utopia point detection. It may be symmetric or non-symmetric, in the symmetric case a similar number of point are selected around the Pareto-best-solution, the first one will tends to better serve the first optimality criteria while the second one will serve better the second optimality criteria. Those neighbors of the Pareto-best-solution are used as local optimums of next MO-Meta heuristic processing. The neighborhood size, $N3$, should be less than the swarm size, $N1$, of the used meta-heuristic. Finally the Pareto corresponding particles are inserted as local bests for next iterative processing such in Figure 3.

Inputs : PSO_swarm_size: $N1$, Repository-size: $N2$, Number of neighborhoods: $N3$.
Output: $Xl [N3]$ // local best solutions for meta-heuristic (PSO).
Begin
1- Estimate utopia_point()
2- Select best Pareto solution
3- Select $N3$ neighbors of the best pareto solution
4- For $i=1$ to $N3$
 $Xl [i] \leftarrow$ X position corresponding the Pareto Neighborhood.
 // $Xl [i]$ is a local best of particle (i) of the meta-heuristic swarm
End for
End ()

Figure 3. The pareto neighborhood PN

3.4. MOPSO-PN for Clustering

To use MOPSO-PN for clustering, each clustering solution (dataset clusters centers) is defined by

the position of a particle Pos_i which is presented by a matrix of size k rows and d columns, where k is the number of clusters and d is the dimension (the number of features) of the dataset.

Each line vector in this matrix presents a cluster center as follows.

$$\text{Pos}_i = \begin{bmatrix} C_{i1} \\ \vdots \\ C_{ij} \\ \vdots \\ C_{ik} \end{bmatrix}, \quad i = [1, \dots, \text{nParticle}] \quad (17)$$

C_{ij} is a vector of size d , it represents the j -th cluster center of the i -th particle.

The fitness (F) of each particle is represented by a vector of size n (n is the number of objective functions). In the first scenario, there are two objective functions presented by two cluster validity indices (the Silhouette and the overall cluster deviation) for that $n=2$ and by consequence $F=[F_1, F_2]$, while in the second scenario there are three objective functions presented by three cluster validity indices (I-Index, Con-Index and Sym-Index) for that $n=3$ and by consequence $F=[F_1, F_2, F_3]$.

3.4.1. Architecture of MOPSO-PN for clustering

The architecture of MOPSO-PN algorithm for clustering is represented by Figure 4. As input, MOPSO-PN receives the dataset to be processed, the initialization parameters and the objectives functions (The algorithm receives two objective functions in the first scenario and three objective functions in the second scenario). As output, MOPSO-PN provides the number of clusters related to the dataset supplied), dataset centers and clustered dataset.

Figure 4. Architecture of MOPSO-PN clustering algorithm

After defining the input data (Dataset for clustering “**Dataset**” , Maximum Number of Iterations “**MaxIteration**” , Number of Particles “**N1**”, Repository Size “**N2**”(N2=N1/2) , Neighborhood size “**N3**” (N3 ≤ N2) for the Pareto Neighborhood and Initial Parameters (**w**, **c1,c2**)), steps of MOPSO-PN for clustering are detailed in Figure 5, these steps are defined as following:

Step 1 consists in defining the range of values of k: $k \in \{ k_1 \dots k_n \}$; this step is made by:

- 1- Defining the interval of values of $k \in \{ k_{min}, \dots, k_{max} \}$ (given a dataset of size S, K_{min} and K_{max} present respectively the minimum and the maximum number of clusters; the default value of K_{min} is 2; and the default value of K_{max} is $\sqrt{S} + 1$ in the case where it has not been specified) [36]
- 2- For each value of k in this range calculate its corresponding TMD value using PSO based Total-Mean-Distance (TMD) algorithm and store in each time the value of k as well as its corresponding value of TMD in a matrix k rows 2 columns;
- 3- Select the n best values of k with the minimum values of TMD.

Step 2, 3, 4, 5, 6 and 7 are defined as the initialization phase of MOPSO-PN for clustering and step 8 and 9 corresponds respectively to the optimization phase and the decision making phase of MOPSO-PN for clustering. These steps are repeated for each value of k in the range of values of k and in each time a triplet constituted of the value of k, the best value of F-measure and the corresponding optimal solution is stored.

Step 10 consists in selecting the best value ok k with the maximum value of “F-measure” and its corresponding optimal solution to produce as output the best value of k with the best F-measure and the best position of cluster centers (particle’s positions) and consequently the dataset clustered according to the clusters centers.

Input: Dataset, MaxIteration, N1, N2, N3, w, c1, c2.

- 1) Define the range of values of k : $k \in \{k_1 \dots k_n\}$
 For each value of k in $\{k_1 \dots k_n\}$ **do** :
 iter $\leftarrow$ 1; F-measure $\leftarrow$ 0.
 - 2) Generate initial positions of Particles “Particle.Position” (Clustering solutions) and initial velocities “Particle.Velocity”.
 - 3) Calculate the Cost of each Particle: Particle.Fitness $\leftarrow$ FitnessFunction (Particle.Position);
 - 4) Set the initial Personal Best “**PBest**” of each Particle.
 - 5) Determine domination between positions of Particles.
 - 6) Fill the Repository: Repository $\leftarrow$ non-dominated-solutions.
 - 7) Define the “Utopia-Point” in the Repository and select the “Best-Solution” as the closest point to the “Utopia-Point”.
 - 8) **While** (iter < MaxIteration+1) && (F-measure $\neq$ 1)
 GBest $\leftarrow$ Best-Solution
 6.1) Apply the Pareto Neighborhood topology.
 6.2) Update velocity and position of each Particle as in Equations (15) and (16).
 6.3) Calculate the new Cost of each Particle.
 6.4) Update the Particle Personal Best (**PBest**).
 6.5) Determine domination between Particles and fill the Repository with Non-Dominated Particles.
 6.6) Determine domination of new Repository members.
 6.7) Keep only Non-Dominated members in the Repository.
 6.8) Check if Repository is full and keep only N2 members
 { **If** Size (Rep) > N2 //the current size of the Repository exceeds its initial size
 Define the “Utopia-Point” in the Repository;
 Rep $\leftarrow$ N2 solutions selected from Rep as the closest points to “Utopia-Point”;
 Else
 // Do Nothing
 End If
 6.9) Select the final “Best-Solution” (Best clustering solution related to this iteration), from the Repository, as the closest point to the “Utopia-Point”.
 6.10) Calculate the “F-measure” (obtained after the classification of the dataset based on “Best-Solution”) $\rightarrow$ In each iteration this “Best-Solution” and its corresponding “F-measure” are stored.
 iter $\leftarrow$ iter+1.
 End While
 - 9) Select the “Best-Solution” with the maximum “F-measure” from the set of stored “Best-Solution” and “F-measure” obtained in the end of all iterations.
 End For
 - 10) Select the best value ok k with the maximum value of “F-measure” and its corresponding “Best-Solution”.
- Output:** number of cluster k + “Best-Solution” having the maximum “F-measure”.
-

Figure 5. Pseudo code of MOPSO-PN clustering algorithm

4. EXPERIMENTAL INVESTIGATIONS

This section contains an evaluation of the performances of MOPSO-PN clustering algorithm over 9 real-life datasets and 14 artificial datasets and a comparison with state of art algorithms.

4.1. Processing environment and Dataset

MOPSO-PN has been implemented in Matlab R2018b on an intel core i7 7th Gen, with 2.9 GHz, 8 GB RAM in a Windows 10 environment. All tests are organized in two scenarios. In the first scenario, MOPSO-PN processed 3 real-life datasets (Votes [20], Soybean [20] and Zoo [20]). In the second scenario it processed 6 real-life datasets (Iris [20], Cancer [20], Newthyroid [20], Wine [20], LiverDisorder [20] and Glass [20]) and 14 artificial datasets (Sph-5-2 [2], Sph-4-3 [2], Sph-6-2 [2], Sph-10-2 [3], Sph-9-2 [3], Pat 1 [28], Pat 2 [28], Long 1 [14], Sizes 5 [14] , Spiral [14] , Square1 [14], Square4 [14] , Twenty [14] and Fourty [14]. Table 2 defines the number of instances (Points), number of features (Dimension d) and number of clusters (k) related to each one of these datasets.

Table 2. Description of datasets.

	Dataset	# Points	Dimension (d)	# Clusters
Real-life datasets	Votes	435	16	2
	Soybean	47	35	4
	Zoo	101	16	7
	Iris	150	4	3
	Cancer	683	9	2
	Newthyroid	215	5	3
	LiverDisorder	345	6	2
	Glass	214	9	6
	Wine	178	13	3
Artificial datasets	Sph-5-2	250	2	5
	Sph-4-3	400	3	4
	Sph-6-2	300	2	6
	Sph-10-2	500	2	10
	Sph-9-2	900	2	9
	Pat 1	557	2	3
	Pat 2	417	2	2
	Long 1	1000	2	2
	Sizes 5	1000	2	4
	Spiral	1000	2	2

Square 1	1000	2	4
Square 4	1000	2	4
Twenty	1000	2	20
Fourty	1000	2	40

4.2 Test Scenarios

To evaluate the performances of MOPSO-PN, two scenarios have been developed and detailed in the next paragraphs.

4.2.1 First test scenario

In this scenario, MOPSO-PN algorithm used two cluster validity indices as objectives functions (the Silhouette and the overall cluster deviation $Dev(C)$) and three real-life datasets (Votes, Zoo and Soybean) for the test. MOPSO-PN algorithm was executed 30 times with the following input parameters showed in Table 3.

Table 3. MOPSO-PN parameters in Scenario 1

Parameter	Value
Number-particles	50
Repository-size	25
Number-iterations	100
W	0.8
C1	1.42
C2	1.63

The metric used to evaluate the performances of MOPSO-PN algorithm is the Minkowski Score (MS) [19] [29]; The interval of values of MS is [0,1]; Lower values of MS imply better clustering results. The obtained results were compared with those given by Mukhopadhyay & Maulik [26] and Cunha et al [8].

The clustering algorithms used for comparison are: SGA [10], MOGA [11] and cOptBees-MO [8].

4.2.2 Second test scenario

In this scenario, MOPSO-PN algorithm used three cluster validity indices as objectives functions (the I-index, the Sym-index and the Con-index. 6 real-life datasets (Iris, Cancer, Newthyroid, Liver Disorder, Glass and Wine) and 14 artificial datasets (Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2,

Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty and Fourty) are used for the test.

MOPSO-PN was executed 30 times with the same parameters used in the first scenario. The metric used for evaluation is the F-measure [31] [13]; The interval of values of F-measure is [0,1]; Higher values of F-Measure imply better clustering results. The obtained results were compared with those given by [8] [1] [37] [21] [35] [33]. The clustering algorithms used for comparison are: MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, VGAPS, cOptBee-MO, HCHPS-MOEC, RVMOC, MOCSO and MOQMFO.

4.3 RESULTS AND DISCUSSIONS

In this section we will present and discuss the results provided by MOPSO-PN.

4.3.1 Results and discussions of the first scenario

The Average Minkowski scores of MOPSO-PN and associated algorithms from the literature over Votes, Zoo and Soybean datasets are showed in Table 4. Bests results (with lower values) are marked in bold. Here, we distinguish two version of cOptBees-MO: cOptBees-MO1 and cOptBees-MO2. Provided results in Table 4 show that, although cOptBees-MO in its second version, which optimizes the silhouette-index and number of clusters, gives the best results for all datasets, it does not gives good results in its first version which optimizes the silhouette-index and dev(c), but using those last two indices, MOPSO-PN performs well.

Table 4. for the SGA, MOGA, cOptBees-MO, MOPSO-PN over the Votes, Zoo and Soybean datasets (best values of Average-Minkowski-score are marked in bold)

Dataset	SGA	MOGA	cOptBees-MO 1	cOptBees-MO 2	MOPSO-PN
Votes	0.6550	0.5344	0.6290	0.164	0.3126
Zoo	0.5096	0.3807	0.6679	0.000	0.2566
Soybean	0.4147	0.0000	0.6604	0.000	0.0000

4.3.2. Results and discussions of the second scenario

The MOPSO-PN algorithm was applied with 6 real-life datasets and 14 artificial datasets. Comparisons are based on F-measure values and number of clusters (k) obtained with MOPSO-PN and its competitors. Table 5 shows F-measure values and numbers of clusters provided by each algorithm. Bests results (with higher values) are marked in bold.

These results illustrated in Table 5 prove that MOPSO-PN had competitive performances when compared with other algorithms and was able to detect the real number of cluster of all datasets.

MOPSO-PN gave the best values of F-Measure with the following datasets: Iris (F-Measure=1), Newthyroid (F-Measure=1), Wine (F-Measure=0.93), Sph-5-2 (F-Measure=0.99), Sph-4-3 (F-Measure=1), Sph-6-2 (F-Measure=1), Sph-10-2 (F-Measure=1), Sph-9-2 (F-Measure=0.92),

Long 1 (F-Measure=1),Size 5 (F-Measure=0.99), Square 1 (F-Measure=0.99), Twenty (F-Measure=1), and Fourty (F-Measure=1).

However MOPSO-PN provided acceptable values of F-Measure for the following cases: Cancer (F-Measure=0.91), LiverDisorder (F-Measure=0.72), Glass (F-Measure=0.66), Pat 1 (F-Measure=0.60), Pat 2 (F-Measure=0.82), Spiral (F-Measure=0.66), Square 4 (F-Measure=0.93).

The best value of F-Measure for Glass dataset was provided by cOptBeeMO with value of 0.88 followed by MOPSO-PN with value of 0.66, but cOptBeeMO unlike MOPSO-PN failed in detecting the real number of clusters.

The clustering results of the artificial datasets based on MOPSO-PN Clustering algorithm are presented in Figure 6: Sph-5-2 presented in Figure 6a, Sph-4-3 illustrated in Figure 6b, Sph-6-2 showed in Figure 6c, Sph-10-2 presented in Figure 6d, Sph-9-2 illustrated in Figure 6e, Pat 1 depicted in Figure 6f, Pat 2 illustrated in Figure 6g, Long 1 showed in Figure 6h, Sizes 5 presented in Figure 6i, Spiral showed in Figure 6j, Square 1 illustrated in Figure 6k, Square 4 depicted in Figure 6l, Twenty presented in Figure 6m and Fourty illustrated in Figure 6n.

Results showed in Table 6 are based on 30 generations, such in MOPSOSA, VGAPS and MOQMFO. MOPSO-PN provided the best values of average F-measure and standard deviation for Iris, Newthyroid, Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Long 1, Sizes 5, Square 1, Twenty and Fourty.

Table 5. F-measure and number of clusters obtained using MOPSO-PN and its competitors (best values of F-measure are marked in bold)

Dataset	#Clusters	MOPSOSA	GenClustMOO	GenClustPESA2	MOCK	VGAPS	cOptBee-MO	MOCSSO	MOQMFO	MOPSO-PN
		k F- Measure	k F-Measure	k F- Measure	k F- Measure	k F- Measure	k F- Measure	F- Measure	k F-Measure	k F-Measure
Iris	3	3 0.92	3 0.79	3 0.93	2 0.78	3 0.76	3 0.86	0.95	3 1.00	3 1.00
Cancer	2	2 0.98	2 0.97	2 0.98	2 0.82	2 0.95	3 0.94	-	2 0.91	2 0.90
Newthyroid	3	3 0.89	3 0.86	9 0.69	2 0.74	5 0.66	4 0.86	-	3 0.96	3 1.00
LiverDisorder	2	2 0.69	2 0.67	5 0.60	2 0.67	2 0.70	2 0.67	-	2 0.75	2 0.72
Glass	6	6 0.57	6 0.49	5 0.53	5 0.53	5 0.53	2 0.88	-	6 0.60	6 0.66
Wine	3	- -	3 0.71	13 0.44	3 0.73	6 0.62	2 0.65	0.73	3 0.91	3 0.93
Sph-5-2	5	5 0.98	5 0.97	5 0.94	6 0.91	5 0.55	- -	-	5 0.96	5 0.99
Sph-4-3	4	4 1.00	4 1.00	4 1.00	4 1.00	4 1.00	- -	-	4 1.00	4 1.00
Sph-6-2	6	6 1.00	6 1.00	6 1.00	6 1.00	6 1.00	- -	-	6 1.00	6 1.00
Sph-10-2	10	10 0.99	10 0.99	12 0.94	6 0.72	7 0.76	- -	-	10 0.96	10 1.00
Sph-9-2	9	9 0.92	9 0.69	8 0.66	9 0.73	9 0.49	- -	-	9 0.80	9 0.92
Pat 1	3	3 1.00	3 0.95	3 0.95	10 0.55	4 0.42	- -	-	3 0.77	3 0.60
Pat 2	2	2 1.00	2 1.00	2 1.00	11 0.55	4 0.59	- -	-	2 0.82	2 0.82
Long 1	2	2 1.00	2 1.00	2 1.00	2 1.00	3 0.50	- -	0.99	2 1.00	2 1.00
Sizes 5	4	4 0.98	4 0.97	3 0.88	2 0.80	5 0.82	- -	0.98	4 0.99	4 0.99
Spiral	2	2 1.00	2 1.00	2 1.00	3 0.95	6 0.38	- -	-	2 0.65	2 0.66
Square 1	4	4 0.99	4 0.99	4 0.99	4 0.99	4 0.99	- -	-	4 0.97	4 0.99
Square 4	4	4 0.94	4 0.92	4 0.88	4 0.90	2 0.93	- -	-	4 0.92	4 0.93
Twenty	20	20 1.00	20 1.00	24 0.95	20 1.00	20 0.48	- -	-	20 1.00	20 1.00
Fourty	40	40 1.00	40 1.00	40 0.98	40 1.00	2 0.10	- -	-	40 1.00	40 1.00

Table 6. Average F-measure and standard deviation obtained using MOPSO-PN and its competitors (best values of Average F-measure are marked in bold)

Dataset	MOPSOSA	GenClustMOO	GenClustPESA2	MOCK	VGAPS	cOptBee-MO	HCHPS-MOEC	RVMOC	MOC SO	MOQMFO	MOPSO-PN
Iris	0.93 ± 0.011	0.78 ± 0.011	0.92 ± 0.015	0.77 ± 0.022	0.75 ± 0.013	0.83 ± 0.024	1.00 ± 0.000	0.82 ± 0.0769	0.94	0.96 ± 0.035	1.00 ± 0.000
Cancer	0.98 ± 0.003	0.96 ± 0.009	0.97 ± 0.014	0.91 ± 0.014	0.95 ± 0.012	0.91 ± 0.021	0.97 ± 0.010	0.84± 0.0000	-	0.86 ± 0.048	0.89 ± 0.012
Newthyroid	0.88 ± 0.010	0.86 ± 0.016	0.68 ± 0.015	0.73 ± 0.014	0.65 ± 0.011	0.83 ± 0.021	0.92 ± 0.011	0.49 ± 0.0198	-	0.84 ± 0.062	0.99 ± 0.013
LiverDisorder	0.77 ± 0.010	0.67 ± 0.002	0.60 ± 0.015	0.67 ± 0.012	0.70 ± 0.009	0.61 ± 0.050	-	-	-	0.71 ± 0.035	0.71 ± 0.015
Glass	0.56 ± 0.002	0.49 ± 0.012	0.53 ± 0.012	0.53 ± 0.006	0.53 ± 0.008	0.79 ± 0.040	0.54 ± 0.011	-	-	0.55 ± 0.027	0.65 ± 0.002
Wine	-	0.70 ± 0.012	0.43 ± 0.012	0.72 ± 0.002	0.61 ± 0.008	0.64 ± 0.007	0.95 ± 0.009	0.95± 0.0019	0.73	0.82 ± 0.055	0.91 ± 0.031
Sph-5-2	0.98 ± 0.006	0.95 ± 0.021	0.93 ± 0.012	0.90 ± 0.011	0.54 ± 0.011	-	-	-	-	0.88 ± 0.049	0.98 ± 0.017
Sph-4-3	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	-	-	-	-	1.00 ± 0.000	1.00 ± 0.000
Sph-6-2	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	-	-	-	-	1.00 ± 0.000	1.00 ± 0.000
Sph-10-2	0.99 ± 0.002	0.98 ± 0.011	0.93 ± 0.021	0.71 ± 0.013	0.75 ± 0.011	-	-	-	-	0.92 ± 0.028	0.99 ± 0.005
Sph-9-2	0.92 ± 0.001	0.68 ± 0.012	0.65 ± 0.018	0.71 ± 0.009	0.48 ± 0.012	-	-	-	-	0.74 ± 0.034	0.92 ± 0.003
Pat 1	0.98 ± 0.012	0.94 ± 0.013	0.94 ± 0.009	0.54 ± 0.011	0.41 ± 0.014	-	-	-	-	0.69 ± 0.049	0.59± 0.010
Pat 2	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	0.54 ± 0.013	0.58 ± 0.021	-	-	-	-	0.80 ± 0.017	0.81 ± 0.003
Long 1	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	0.48 ± 0.021	-	-	-	0.99	1.00 ± 0.000	1.00 ± 0.000
Sizes 5	0.97 ± 0.001	0.96 ± 0.001	0.88 ± 0.011	0.79 ± 0.012	0.81 ± 0.013	-	-	-	0.98	0.93 ± 0.059	0.99 ± 0.002
Spiral	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	0.94 ± 0.011	0.37 ± 0.016	-	1.00 ± 0.000	-	-	0.62 ± 0.019	0.65 ± 0.002
Square 1	0.99 ± 0.011	0.99 ± 0.013	0.99 ± 0.014	0.99 ± 0.012	0.99 ± 0.014	-	-	-	-	0.90 ± 0.060	0.99 ± 0.001
Square 4	0.93 ± 0.001	0.91 ± 0.014	0.87 ± 0.011	0.89 ± 0.011	0.92 ± 0.013	-	-	-	-	0.78 ± 0.059	0.92 ± 0.011
Twenty	1.00 ± 0.000	1.00 ± 0.000	0.94 ± 0.015	1.00 ± 0.000	0.47 ± 0.022	-	-	-	-	1.00 ± 0.000	1.00 ± 0.000
Fourty	1.00 ± 0.000	1.00 ± 0.000	0.97 ± 0.015	1.00 ± 0.000	0.95 ± 0.006	-	-	-	-	1.00 ± 0.000	1.00 ± 0.000

Figure 6. Graphical representation of used artificial-datasets after the application of MOMFO-PN technique.

5. CONCLUSIONS AND PERSPECTIVES

In this paper a new topology of Pareto neighborhood is integrated to MOPSO to produce MOPSO-PN, and this in order to find high quality clustering solutions. The study is made based on two scenarios: In the first scenario MOPSO-PN used two cluster validity criteria (the Silhouette and the Overall cluster deviation (Dev(C))) as objective functions, three datasets (Votes, Zoo and Soybean) and was compared to (SGA, MOGA and cOptBees-MO) based on Average Minkowski-Score metric; In the second scenario MOPSO-PN used three cluster validity criteria (the I-index, the Sym-index and the Con-index) as objective functions, 20 datasets (Iris, Cancer, Newthyroid, Liver Disorder, Glass, Wine, Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Pat1, Pat2, Long1, Sizes5, Spiral, Square1, Square4, Twenty, Forty) and was compared to (MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, VGAPS, cOptBee-MO, HCHPS-MOEC, RVMOC, MOCSO and MOQMFO) basing on the number of clusters detected, F-measure, Average F-measure, and Standard Deviation metric.

In both scenarios, MOPSO-PN proved its capability to produce a good result of clustering solutions for almost datasets and was able to find the correct number of clusters. Best results are observed in: Soybean, Iris, Newthyroid, Wine, Sph-5-2, Sph-4-3, Sph-6-2, Sph-10-2, Sph-9-2, Long 1, Sizes 5, Square 1, Twenty, Forty.

As future work we will focus on ameliorating the performances of MOPSO-PN over datasets with higher values of Minkowski Score or less values of F-measure.

6. REFERENCES

- [1] Abubaker, A., Baharum, A. & Alrefaei, M., 2015. Automatic clustering using multi-objective particle swarm and simulated annealing. *PloS one*, 10(7).
- [2] Bandyopadhyay, S. & Maulik, U., 2002. Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern recognition*, 35(6), pp.1197-208.
- [3] Bandyopadhyay, S. & Pal, S.K., 2007. *Classification and learning using genetic algorithms: applications in bioinformatics and web intelligence*. Springer Science & Business Media..
- [4] Bandyopadhyay, S. & Saha, S., 2007. GAPS: A clustering method using a new point symmetry-based distance measure. *Pattern recognition*, 40(12), pp.3430-51.
- [5] Bandyopadhyay, S. & Saha, S., 2008. A point symmetry-based clustering technique for automatic evolution of clusters. *IEEE Transactions on Knowledge and Data Engineering*, 20(11), pp.1441-57.

- [6] Coello, C.A.C., Lamont, G.B. & Van, V., 2007. Evolutionary algorithms for solving multi-objective problems. 5(79-104).
- [7] Coello, C.C. & Lechuga, M.S., (2002, May). MOPSO: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600).IEEE, Vol. 2, pp.1051-56.
- [8] Cunha, D. et al., 2017. Bio-inspired multiobjective clustering optimization: A survey and a proposal. Artificial Intelligence Research, 6(2), pp.10-24.
- [9] Deb, K., 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In Multi-objective evolutionary optimisation for product design and manufacturing. pp.3-34.
- [10] Dhiraj, K. & Rath, S.K., 2009. Comparison of SGA and RGA based clustering algorithm for pattern recognition. International Journal of Recent Trends in Engineering, 1(269).
- [11] Dutta, D., Dutta, P. & Sil, J., 2012. Simultaneous feature selection and clustering for categorical features using multi objective genetic algorithm. 12th international conference on hybrid intelligent systems (HIS), pp.191-96.
- [12] Freitas, A.A., 2004. A critical review of multi-objective optimization in data mining: a position paper. ACM SIGKDD Explorations Newsletter, 6(2), pp.77-86.
- [13] Fung, B.C., Wang, K. & Ester, M., 2003. Hierarchical document clustering using frequent itemsets. Proceedings of the 2003 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, pp.59-70.
- [14] Handl, J. & Knowles, J., 2007. An evolutionary approach to multiobjective clustering. IEEE transactions on Evolutionary Computation, 11(1), pp.56-76.
- [15] Hironaka, A., Okamoto, T., Koakutsu, S. & Hirata, 2015. Analysis and improvements of the Pareto optimal solution visualization method using the self-organizing maps. SICE Journal of Control, Measurement, and System Integration, pp.34-43.
- [16] Hossein, S.S., Hamidi, S.A., Mansuri, M. & Ghoddosian, A., 2015. Multi Objective Particle Swarm Optimization (MOPSO) for size and shape optimization of 2D truss structures. Periodica Polytechnica Civil Engineering, 59(1), pp.9-14.
- [17] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), pp.651-66.
- [18] Jain, A.K. & Dubes, R.C., 1988. Algorithms for clustering data. Prentice-Hall, Inc.
- [19] Jardine, N. & Sibson, R., 1971. Mathematical taxonomy. (512.81 J3).
- [20] Lichman, M., 2013. UCI machine learning repository.

- [21]Liu, C., Li, Y., Zhao, Q. & Liu, C., 2019. Reference vector-based multi-objective clustering for high-dimensional data. *Applied Soft Computing*, 78, pp.614-29.
- [22]Matake, N., Hiroyasu, T., Miki, M. & Senda, T., 2007. Multiobjective clustering with automatic k-determination for large-scale data. *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp.861-68.
- [23]Maulik, U. & Bandyopadhyay, S., 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on pattern analysis and machine intelligence*, 24(12), pp.1650-54.
- [24]Maulik, U., Bandyopadhyay, S. & Mukhopadhyay, A., 2011. *Multiobjective genetic algorithms for clustering: applications in data mining and bioinformatics*. Springer Science & Business Media.
- [25]Monti, S., Tamayo, P., Mesirov, J. & Golub, T., 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1), pp.91-118.
- [26]Mukhopadhyay, A. & Maulik, U., 2007. Multiobjective approach to categorical data clustering. *IEEE Congress on Evolutionary Computation*, pp.1296-303.
- [27]Mukhopadhyay, A., Maulik, U. & Bandyopadhyay, S., 2015. A Survey of Multiobjective Evolutionary Clustering. *ACM computing surveys*, 47(4), pp.1-46.
- [28]Pal, S.K. & Mitra, S., 1994. Fuzzy versions of Kohonen's net and MLP-based classification: performance evaluation for certain nonconvex decision regions. *Information Sciences*, 76(3-4), pp.297-337.
- [29]Saha, S. & Bandyopadhyay, S., 2010. A symmetry based multiobjective clustering technique for automatic evolution of clusters. *Pattern recognition*, 43(3), pp.738-51.
- [30]Saha, S. & Bandyopadhyay, S., 2012. Some connectivity based cluster validity indices. *Applied Soft Computing*, 12(5), pp.1555-65.
- [31]Saha, S. & Bandyopadhyay, S., 2013. A generalized automatic clustering algorithm in a multiobjective framework. *Applied Soft Computing*, 13(1), pp.89-108.
- [32]Shi, y. & Eberhart, R., 1998. A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings.*, pp.69-73.
- [33]Soussi, Y., Rokbani, N., Wali, A. & Alimi, A., 2020. Multi-objective Quantum Moth Flame Optimization for Clustering.
- [34]Tsai, D.M. & Lin, C.C., 2011. Fuzzy C-means based clustering for linearly and nonlinearly separable data. *Pattern recognition*, 44(8), pp.1750-60.

- [35] Yan, D. et al., 2020. Single-objective/multiobjective cat swarm optimization clustering analysis for data partition. *IEEE Transactions on Automation Science and Engineering*, 17(3), pp.1633-46.
- [36] Yang, S.L., Li, Y.S., Hu, X.X. & Pan, R.Y., 2006. Optimization Study on k Value of K-means Algorithm. *Systems Engineering-Theory & Practice*, 2, pp.97-101.
- [37] Yan, S. et al., 2019. Hybrid chain-hypergraph P systems for multiobjective ensemble clustering. *IEEE Access*, pp.143511-23.