Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Subspace Discovery through Evolutionary Multi-objective Optimization in Overlapping Clustering

[1]Ramesh marpu, [2]Dr. Bairam Manjula

[1]Research scholar, Bir Tikendrajit University
[2]Research Supervisor, Bir Tikendrajit University

Abstract: This paper employs a multi-objective optimization technique to concurrently partition data into multiple overlapping subspace clusters. Simultaneously, the data grouping and the identification of relevant subspace feature sets corresponding to these groups are performed. The study utilizes validity indices, including the ICC-index, PSM-index, and a novel MNR-index, where the latter optimizes the overlapping of objects into distinct clusters. Furthermore, existing mutation operators such as large deletions, large duplications, and large translocations are adapted to enhance the exploration of the search space effectively. The proposed method is tested on ten standard real-life datasets and sixteen synthetic datasets to identify diverse overlapping subspace clusters. Comparative analyses with existing methods highlight the advantages of incorporating multiple objectives and the newly defined objective function, demonstrating superior performance in the majority of cases. Additionally, the paper illustrates the application of this method in the bi-clustering of gene expression profile data, showcasing its versatility and efficacy across different domains.
Key words: ICC-index, PSM-index, MNR-index, MOO, overlapping, subspace

1 Introduction

Subspace clustering methods aim to categorize a dataset into sets of objects that may overlap or remain distinct. However, when dealing with many features representing each object, it is common to encounter irrelevant or redundant features. To tackle this problem, the suggested method picks unique sets of features for different groups of objects. In subspace clustering, an object can be associated with multiple sets of features, so it's not uncommon for an object to be part of more than one cluster. To address this challenge, an overlapping subspace clustering technique is introduced, allowing objects to be part of multiple clusters.

In unsupervised ML, the idea of optimization [1] is commonly used to solve various real-world issues. The suggested method follows the MOO framework and uses evolutionary techniques to create overlapping clusters in subspaces. To achieve high-quality subspace clusters, optimization is essential, focusing on both cluster compactness and the selection of subspace feature sets. Furthermore, to create overlapping clusters, optimization is necessary to manage the overlapping of objects [7]. The objective function is designed to limit object overlaps and ensure that only those objects relevant to multiple subspace feature sets are included in more than one cluster.

The developed method is an evolutionary approach where we create a genotype, also known as a genome, made up of a set of tuples. Each tuple is denoted as $= \{\tau_1, \tau_2, \ldots \tau_t\}$. In this representation, 'τ_i' signifies a functional element, while '$< h_i, k_i, x_i >$' signifies a non-functional one. In each tuple, hi signifies the quantity of cluster, and fi denotes number of the feature selected. ki and fi are uniformly chosen from $\{1,2, \ldots SC_{max}\}$ and $\{1,2, \ldots, F\}$ their respective SC_{max} limits, k: upper limit for the number of clusters. Again, x ∈ Coord, with $Coord = \left\{ j \times \frac{x_{max}}{1000} \mid j \in \{-1000, \ldots, 1000\} \right\}, x_{max}$ is the highest associated with every feature in normalized dataset.

A phenotype, denoted as P^{θ}, includes a collection of fundamental points. The genotype, G^{θ}, is then translated into a phenotype to establish the locations of core points or centers for subspace clusters. The clusters, θ_i, created by considering the tuples in the genotype where $i \in \{1, 2, \ldots, K\}$. The phenotype is shown as a matrix having a size of $SC_{max} * _{\circ}$. The clusters get reorganized by modifying the tuples found in the G^{θ}. Just the practical tuples, $\forall_i \ h_i = 1$, contribute to updating a phenotype. Inefficient components need to pause till they progress into practical through mutation operators. The components k_i and f_i of a tuple, representing number of the cluster and number of the feature, and their matching location (k^{th} row, f^{th} column) in the phenotype, are revised with the value x_i from that tuple.

The G^{θ} in eq. illustrates how the genotype maps to the phenotype. By a specific instant, with a genome dimension of t = 8, the largest amount of clusters, is set at is $SC_{max} = 4$, and the feature set comprises $F = \{f_1, f_2, f_3\}$.

$$G^{\theta} = \begin{matrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\\ \tau_5\\ \tau_6\\ \tau_7\\ \tau_8\end{matrix}\begin{bmatrix}0 & 2 & 1 & x_2\\ 0 & 3 & 2 & x_3\\ 1 & 1 & 3 & x_2\\ 1 & 3 & 1 & x_5\\ 1 & 2 & 2 & x_4\\ 1 & 1 & 3 & x_6\\ 1 & 2 & 1 & x_6\\ 1 & 3 & 3 & x_7\end{bmatrix} \quad P^{\theta} = \begin{matrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\end{matrix}\begin{bmatrix}0 & 0 & \frac{x_2+x_6}{2}\\ x_6 & x_4 & 0\\ x_5 & 0 & x_7\\ 0 & 0 & 0\end{bmatrix}$$

The tuples in the genome generate three clusters with centers θ1, θ2 & θ3, along with the associated subspace features: F1, F2 and F3. In this approach, the initial genome size is set at 100, with one-third of them being non-functional.

The exploration of the search space involves the application of mutation operators to existing genomes, generating novel genomes. The choice of the most promising genomes for the next steps relies on their objective values, expressed as validity indices.
In this scenario, a novel weight index called the MNR is introduced to measure how much overlap exists among multiple subspace clusters. This is accompanied by further weight indices like the ICC and PSM, each evaluating various aspects of subspace cluster quality. All discussed objectives are simultaneously improved through a multi-objective optimization (MOO) framework. The algorithm we created underwent thorough testing, using seven real-world datasets and sixteen synthetic datasets. We assessed the effectiveness of MOO technique by comparing it to SOO. The evaluation of our method on two categorical datasets and three large datasets demonstrated that our approach is not only competitive but often outperforms state-of-the-art methods.

II. Proposed Methodology

In this section, we have explored various mutation operators employed to traverse the exploring space and objective functions utilized to optimize the resultant clusters.

(a) Mutation Operators

This method includes 2 mutation operators: large-scale reorganization and replacement. The mutation operator of large-scale reorganization involves extensive removal, extensive replication, and extensive displacement. The mutation operators applied to a genome T are as follows:

(b) Point Substitution

Point mutation specifically targets an individual tuple which changes an section inside particular tuple. Mutation operator of substitution uniformly chooses a tuple τi ∈ T and swaps the nth element.

$$\tau_i \leftarrow \begin{Bmatrix} < \rho(\{0,1\}), b, c, d >, for\ = 1 \\ < a, \rho(\{1,2, \dots SC_{max}\}), c, d >, for\ n = 2 \\ < a, b, \rho(\{1,2, \dots, F\}), d >, for\ n = 3 \\ < a, b, c, \rho(\{Coord\}), >, for\ n = 4 \end{Bmatrix}$$

The ρ operator equally picks a component from the scale of respective tuple and substitutes it.

(c) Merge Operator $\widetilde{M}$

For instance, dual tuples represented as $\tau_i =< a, b, c, d > and \tau_j =< a', b', c', d' > andanumberm \in$ {1,2,3,4} is selected consistently. The $\widetilde{M}$ can be illustrated as.

$$\widetilde{M}(\tau_i, \tau_j, m) = \begin{Bmatrix} < a, b', c', d' >, & for\ n = 1 \\ < a, b, c', d' >, & for\ n = 2 \\ < a, b, c, d' >, & for\ n = 3 \\ < a, b, c, d >, & for\ n = 4 \end{Bmatrix}$$

Rearrangement operators influence the size of the genome, leading to either an increase or decrease. These breakpoints within a tuple are reassembled to create new tuples. There are 3 types of operators for this enhancement:

Extensive removal, extensive replication, and extensive displacement involve selecting 2 integers, i and j, evenly inside extent of the genome size. Once again, a discontinuity factor is evenly selected to determine where the change bound is set surrounded by the adjoining tuples. The 3 types of reorganization operators are described as:

(d) Extensive removal

If i less than or equal to j, the portion between the tuples τ_i and τ_j is deleted. If j is less than i, the deletion is carried out with consideration for the circular nature of the genome.

$$T \leftarrow \begin{Bmatrix} T_{1,i-1} + \widetilde{M}(\tau_i, \tau_j, n) + \tau_{j+1,t,} & for\ i \leq j \\ T_{j+1,i-1} + \widetilde{M}(\tau_i, \tau_j, n), & for\ i > j \end{Bmatrix}$$

(e) Extensive replication

If $i \leq j$, the portion among tuples τi and τj is replicated from the primary tuple T and included at a casually selected 3rd position, p. If j<i, the portion is reproduced, respecting the circular nature of the genome.

$$T \leftarrow \begin{Bmatrix} T_{I,p-1} + \widetilde{M}(\tau_p, \tau_i, n) + T_{i+1.j-1} + \\ \widetilde{M}(\tau_j, \tau_p, n) + T_{p+1,t,} & for\ i \leq j \\ T_{I,p-1} + \widetilde{M}(\tau_p, \tau_j, n) + T_{i+1,t} + T_{1,j-1} + \\ \widetilde{M}(\tau_i, \tau_p, n) + T_{p+1,t} & for\ i > j \end{Bmatrix}$$

(f) Extensive Displacement

The portion among tuples τi and τj is removed after the prime tuple T and placed at a casually selected third position, p. For cases where i is less than or equal to j, p 6∈ [i, j] if $j >= i$ and p 6∈ [1, j] ∪ [i, t] if $j < i$. After applying these relocation operatives, the step replacement factor is then employed to the revised genome. The quantity of point substitutions to implement is established through the utilization of the binomial distribution, β (η_r, t′), in which t' is order of the genome after relocations and η_r is the rate of mutation. Still, to prevent order of the genome from becoming zero, the limits of i to j restricted to 1/ 3 rd of the order. The mutation rate η_r is set at 0.005.

III. Objective Functions

Objective functions serve the purpose of minimizing or maximizing certain criteria, and they can also be formulated as validity indices. Consequently, during each step, validity indices are progressively optimized, leading to an optimal solution. The method employs non-conquered ordering and crowding distance, to arrange the results depending on real values and select the best solutions.

The objective values for the subspace are computed once the dataset X is divided into K subspace clusters. Each subspace cluster, denoted as Ci, is characterized by a subspace feature set Fi, where i ranges from 1 to K. The validity indices considered in this approach consist of the ICC index, the PSM index and a newly introduced MNR index. The MNR is defined to address the overlap of objects in overlapping subspace clustering.

In overlapping subspace clustering, this objective function aims to reduce substantial object overlap by curtailing the count of overlying points [2]. If μ_i and μ_j represent the membership degrees of objects i and j correspondingly, the MNR can be described as follows:

$$MNR(X, P^{\theta}) = \frac{1}{kC_2} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \left|\mu_i \cap \mu_j\right|; \forall \mu_i, \mu_j = 1 \ldots\ldots\ldots\ldots (4.1)$$

The term KC2 represents the grouping of 2 clusters selected from K clusters, and | . | denotes the size of the set.

IV. Production of Overlapping Subspace Clusters

The P^{θ} includes clusters along with their respective subspace. The main goal is to allocate every single thing to one or more clusters, as the proposed method permits object overlap, allowing a particular object to be portion of many clusters.

This assignment process is denoted by a membership matrix, μ, where the size is equal to (clusters, objects). The values within μ are either '0' or '1', indicating whether the i-th object is present ('1') or absent ('0') in the kth cluster. In this methodology, every objective is allocated to a minimum of one cluster. If a specific line has multiple '1's, it means the object belongs to multiple clusters. The suggested process utilizes a distance-based approximate to assign objects to clusters, calculating the distance between any two points according to Equation 4.1.

Usually, an object is allocated along a specific cluster depending on lowest gap amongst them. However, the proposed method introduces a degree of flexibility by allowing an object to be assigned to clusters within 10% of the minimum distance. Therefore, the technique designates an object to clusters that meet the following condition:

$$\mu_{ki} = \begin{Bmatrix} 1, & if\ d\left(x_i, \theta_k\right) \le 1.1 * th = \min_k d\left(x_i, \theta_k\right) \\ 0, & oherwise \end{Bmatrix} \text{.....(4.2)}$$

The suggested method transfers an object to every cluster within 1.5 times of the minimum distance which enables objects associated with several clusters that have practically equal distances. Additionally, experimental observations indicate that the average cluster compactness remains largely consistent, whether the 10% flexibility is applied. Moreover, introducing a 10% flexibility results in a minor quantity of objects being allocated to more groups, particularly those located at nearly equal distances.

V. Data Sets and Implementation Details

The presented algorithm underwent evaluation with 30 actual and synthetic datasets, results were matched to various existing algorithms using different evaluation metrics. The parameter settings applied are consistent with those outlined, except for the number of iterations, which is set to I=400. Furthermore, the mutation rate ηr is established at 0.005, and the initial genome size is set at 100 for real datasets and 200 for synthetic datasets.

VI. Results and Analysis

Here, showcased and analyzed the outcomes achieved by implementing the recommended method on various real and synthetic datasets to evaluate its performance.

(a) Actual Life Data Sets

The ranks of each procedure for different evaluation metrics were established using a method like the one outlined. The least and highest scores from the 20 runs are presented in Table 1. The rankings of the various algorithms are depicted in Fig. 1, leading to the following observations:

- The outcomes in Table 1 have statistical significance and are not merely the result of chance. Welch's t-test [reference 9] was performed on different datasets for various evaluation metrics at a 5% significance level, and all obtained p-values [reference 10] were less than 0.05, indicating statistical significance.
- The proposed algorithm excels, securing the top position in metrics primarily designed to assess subspace clustering quality, namely CE and RNIA (Fig. 1(c) and 1(d)). Moreover, in terms of accuracy, the technique gives competitive results, securing the 2nd position (Fig. 1(b)).

However, the suggested approach holds the 6th position in the Entropy metric (Fig. 1(e)) due to the clusters formed by an algorithm. A higher number of clusters often leads to a better entropy score.

Table 1: the actual-life data sets

	F-Measure		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		Coverage		NumCluster		AvgDim		RunTime	
Dataset Breast	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.68	0.63	0.82	0.78	0.32	0.21	0.67	0.59	0.38	0.31	1	1	20	15	12.04	8.38	6326	6017
CHAMELEOCLUST	0.60	0.51	0.76	0.76	0.23	0.11	0.53	0.25	0.25	0.22	1.0	1.0	8	4	16.75	5.75	339	131
CLIQUE	0.67	0.67	0.71	0.71	0.02	0.02	0.40	0.40	0.26	0.26	1.0	1.0	107	107	1.7	1.7	453	453
DOC	0.73	0.61	0.81	0.76	0.11	0.04	0.84	0.07	0.46	0.27	1.0	0.8	60	6	27.2	2.8	1.00E+06	37515
FIRES	0.49	0.03	0.76	0.76	0.03	0.0	0.05	0.0	1.0	0.01	0.76	0.04	11	1	2.5	1	250	31
INSCY	0.74	0.55	0.77	0.76	0.02	0.0	0.24	0.11	0.60	0.39	0.97	0.74	2038	167	11	4.4	134373	63484
KYMEROCLUST	0.66	0.57	0.79	0.76	0.18	0.14	0.56	0.51	0.31	0.25	1.0	1.0	19	13	12.36	9.26	8	7
MINECLUS	0.78	0.69	0.78	0.76	0.19	0.18	1.0	1.0	0.56	0.37	1.0	1.0	64	32	33	33	40359	29437
P3C	0.63	0.63	0.77	0.77	0.04	0.04	0.19	0.19	0.36	0.36	0.85	0.85	28	28	6.9	6.9	6281	6281
PROCLUS	0.57	0.52	0.80	0.74	0.51	0.11	0.65	0.43	0.32	0.23	0.89	0.69	9	2	24	18	703	141
SCHISM	0.67	0.67	0.75	0.69	0.01	0.01	0.36	0.34	0.35	0.34	1.0	0.99	248	197	2.3	2.2	158749	114609
STATPC	0.41	0.41	0.78	0.78	0.16	0.16	0.33	0.33	0.29	0.29	0.43	0.43	5	5	33	33	5187	4906
SUBCLU	0.68	0.51	0.77	0.67	0.02	0.01	0.54	0.04	0.27	0.24	1.0	0.82	357	5	2	1	5265	16

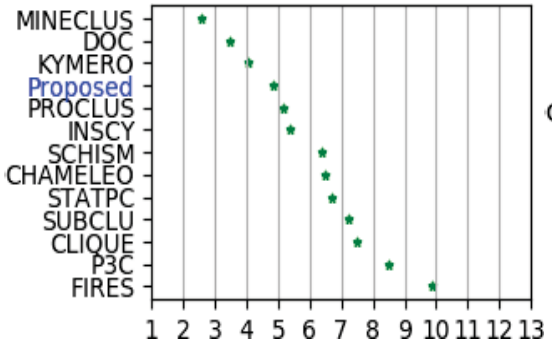
Dataset Diabetes	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.74	0.67	0.77	0.72	0.39	0.28	0.8	0.71	0.29	0.23	1	1	15	10	5.1	4.13	2950	2723
CHAMELEOCLUST	0.70	0.62	0.73	0.70	0.17	0.09	0.66	0.47	0.28	0.23	1.0	1.0	29	19	5	2.75	598	438
CLIQUE	0.70	0.39	0.72	0.69	0.03	0.01	0.14	0.01	0.23	0.13	1.0	1.0	349	202	4.2	2.4	11953	203
DOC	0.71	0.71	0.72	0.69	0.31	0.26	0.92	0.79	0.31	0.24	1.0	0.93	67	17	8	5.1	1.00E+06	51640
FIRES	0.52	0.03	0.65	0.64	0.12	0.0	0.27	0.0	0.68	0.0	0.81	0.03	17	1	2.5	1	4234	360
INSCY	0.65	0.39	0.70	0.65	0.37	0.11	0.45	0.42	0.44	0.15	0.83	0.73	132	3	6.7	5.7	112093	33531
KYMEROCLUST	0.69	0.64	0.73	0.70	0.21	0.08	0.55	0.40	0.25	0.21	1.0	1.0	16	13	3.69	2.87	3	3
MINECLUS	0.72	0.66	0.71	0.69	0.63	0.13	0.89	0.58	0.29	0.17	0.99	0.96	39	3	6	5.2	3578	62
P3C	0.39	0.39	0.66	0.65	0.56	0.11	0.85	0.22	0.09	0.07	0.97	0.88	2	1	7	2	656	141
PROCLUS	0.67	0.61	0.72	0.71	0.34	0.21	0.78	0.69	0.23	0.19	0.92	0.78	9	3	8	6	360	109
SCHISM	0.7	0.62	0.73	0.68	0.08	0.01	0.36	0.09	0.34	0.2	1.0	0.79	270	21	4.2	3.9	35468	250
STATPC	0.73	0.59	0.70	0.65	0.06	0.0	0.63	0.17	0.72	0.28	0.97	0.75	363	27	8	8	27749	4657
SUBCLU	0.74	0.45	0.71	0.68	0.01	0.01	0.01	0.01	0.14	0.11	1.0	1.0	1601	325	4.7	4	190122	58718

Dataset Glass	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.57	0.5	0.65	0.58	0.33	0.26	0.79	0.72	0.52	0.48	1	1	13	7	4.83	3.21	2867	2215
CHAMELEOCLUST	0.43	0.28	0.57	0.50	0.43	0.26	0.88	0.55	0.46	0.36	1.0	1.0	8	4	7.5	4.75	195	95
CLIQUE	0.51	0.31	0.67	0.50	0.02	0.0	0.06	0.0	0.39	0.24	1.0	1.0	6169	175	5.4	3.1	411195	1375
DOC	0.74	0.50	0.63	0.50	0.23	0.13	0.93	0.33	0.72	0.50	0.93	0.91	64	11	9	3.3	23172	78
FIRES	0.30	0.30	0.49	0.49	0.21	0.21	0.45	0.45	0.40	0.40	0.86	0.86	7	7	2.7	2.7	78	78
INSCY	0.57	0.41	0.65	0.47	0.23	0.09	0.54	0.26	0.67	0.47	0.86	0.79	72	30	5.9	2.7	4703	578
KYMEROCLUST	0.65	0.51	0.71	0.60	0.32	0.24	0.85	0.76	0.65	0.55	1.0	1.0	23	19	6.74	5.7	18	16
MINECLUS	0.76	0.40	0.52	0.50	0.24	0.19	0.78	0.45	0.72	0.46	1.0	0.87	64	6	7	4.3	907	15
P3C	0.28	0.23	0.47	0.39	0.14	0.13	0.3	0.27	0.43	0.38	0.89	0.81	3	2	3	3	32	31
PROCLUS	0.60	0.56	0.60	0.57	0.13	0.05	0.51	0.17	0.76	0.68	0.79	0.57	29	26	8	2	375	250
SCHISM	0.46	0.39	0.63	0.47	0.11	0.04	0.33	0.20	0.44	0.38	1.0	0.79	158	30	3.9	2.1	313	31
STATPC	0.75	0.40	0.49	0.36	0.19	0.05	0.67	0.37	0.88	0.36	0.93	0.8	106	27	9	9	1265	390
SUBCLU	0.50	0.45	0.65	0.46	0.0	0.0	0.01	0.01	0.42	0.39	1.0	1.0	1648	831	4.9	4.3	14410	4250

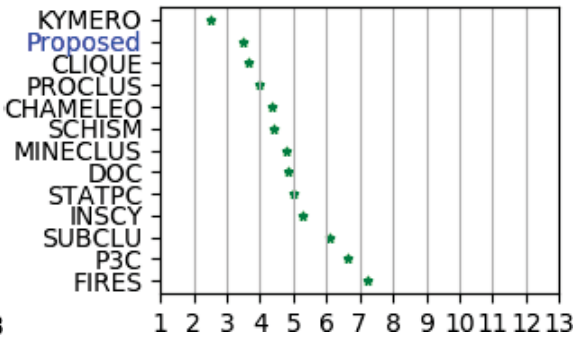
Dataset Liver	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.67	0.6	0.67	0.61	0.39	0.29	0.82	0.71	0.14	0.11	1	1	10	4	3.89	2.73	1551	1350
CHAMELEOCLUST	0.65	0.59	0.68	0.62	0.20	0.10	0.53	0.41	0.14	0.07	1.0	1.0	27	22	2.48	1.85	202	158
CLIQUE	0.68	0.65	0.67	0.58	0.08	0.02	0.38	0.03	0.10	0.02	1.0	1.0	1922	19	4.1	1.7	38281	15
DOC	0.67	0.64	0.68	0.58	0.11	0.07	0.51	0.35	0.18	0.11	0.99	0.9	45	13	3	1.9	625324	1625
FIRES	0.58	0.04	0.58	0.56	0.14	0.0	0.39	0.01	0.37	0.0	0.84	0.03	10	1	3	1	531	46
INSCY	0.66	0.66	0.62	0.61	0.03	0.03	0.42	0.39	0.21	0.20	0.85	0.81	166	130	2.1	2.1	407	234
KYMEROCLUST	0.65	0.56	0.67	0.60	0.21	0.09	0.56	0.43	0.12	0.04	1.0	1.0	17	11	2.82	2	2	2
MINECLUS	0.73	0.63	0.65	0.58	0.09	0.09	0.68	0.48	0.33	0.16	0.99	0.92	64	32	4	3.7	49563	1954
P3C	0.36	0.35	0.58	0.58	0.55	0.27	0.96	0.47	0.02	0.01	0.98	0.94	2	1	6	3	172	32
PROCLUS	0.53	0.39	0.63	0.63	0.26	0.11	0.66	0.25	0.05	0.05	0.83	0.46	6	2	5	3	78	31
SCHISM	0.69	0.69	0.68	0.59	0.04	0.03	0.45	0.26	0.10	0.08	0.99	0.99	90	68	2.7	2.1	31	0
STATPC	0.69	0.57	0.65	0.58	0.23	0.01	0.58	0.37	0.63	0.05	0.77	0.71	159	4	6	3.3	1890	781
SUBCLU	0.68	0.68	0.64	0.58	0.11	0.02	0.68	0.05	0.07	0.02	1.0	1.0	334	64	3.4	1.3	1422	47

Dataset Shape	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.66	0.6	0.79	0.71	0.26	0.21	0.87	0.77	0.72	0.67	1	1	30	23	7.95	4.25	4803	4580
CHAMELEOCLUST	0.75	0.63	0.80	0.71	0.54	0.49	0.78	0.71	0.77	0.67	1.0	1.0	14	10	12.4	10.79	462	252
CLIQUE	0.31	0.31	0.76	0.76	0.01	0.01	0.07	0.07	0.66	0.66	1.0	1.0	486	486	3.3	3.3	235	235
DOC	0.90	0.83	0.79	0.54	0.56	0.38	0.90	0.82	0.93	0.86	1.0	1.0	52	29	13.8	12.8	2.00E+06	86500
FIRES	0.36	0.36	0.51	0.44	0.20	0.13	0.25	0.20	0.88	0.82	0.45	0.39	10	5	7.6	5.3	63	47
INSCY	0.84	0.59	0.76	0.48	0.18	0.16	0.37	0.24	0.94	0.87	0.88	0.82	185	48	9.8	9.5	22578	11531
KYMEROCLUST	0.82	0.72	0.86	0.79	0.57	0.53	0.86	0.80	0.83	0.77	1.0	1.0	19	16	13.5	12.56	101	91
MINECLUS	0.94	0.86	0.79	0.60	0.58	0.46	1.0	1.0	0.93	0.82	1.0	1.0	64	32	17	17	46703	3266
P3C	0.51	0.51	0.61	0.61	0.14	0.14	0.17	0.17	0.8	0.8	0.66	0.66	9	9	4.1	4.1	140	140
PROCLUS	0.84	0.81	0.72	0.71	0.25	0.18	0.61	0.37	0.93	0.91	0.89	0.79	34	34	13	7	593	469
SCHISM	0.51	0.3	0.74	0.49	0.10	0.0	0.26	0.01	0.85	0.55	1.0	0.92	8835	90	6	3.9	712964	9031
STATPC	0.43	0.43	0.74	0.74	0.45	0.45	0.55	0.55	0.56	0.56	0.92	0.92	9	9	17	17	250	171
SUBCLU	0.36	0.29	0.70	0.64	0.0	0.0	0.05	0.04	0.89	0.88	1.0	1.0	3468	3337	4.5	4.1	4063	1891

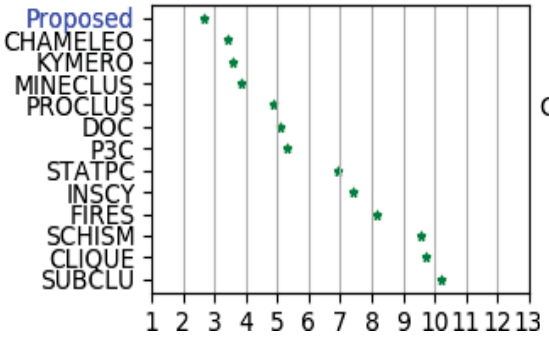
Dataset Vowel	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.39	0.33	0.44	0.38	0.18	0.14	0.87	0.75	0.33	0.27	1	1	32	22	6.66	3.86	11023	8487
CHAMELEOCLUST	0.41	0.37	0.42	0.38	0.17	0.13	0.65	0.54	0.45	0.40	1.0	1.0	33	24	6	4.57	995	787
CLIQUE	0.23	0.17	0.64	0.37	0.05	0.0	0.44	0.01	0.10	0.09	1.0	1.0	3062	267	4.9	1.9	523233	1953
DOC	0.49	0.49	0.44	0.44	0.14	0.14	0.85	0.85	0.58	0.58	0.86	0.86	64	64	10	10	120015	120015
INSCY	0.82	0.33	0.61	0.15	0.09	0.07	0.75	0.26	0.94	0.21	0.90	0.81	163	74	9.5	4.3	75706	39390
FIRES	0.16	0.14	0.13	0.11	0.02	0.02	0.14	0.13	0.16	0.13	0.50	0.45	32	24	2.1	1.9	563	250
MINECLUS	0.48	0.43	0.37	0.37	0.09	0.04	0.62	0.34	0.60	0.46	0.98	0.87	64	64	7.2	3.6	7734	5204
KYMEROCLUST	0.53	0.48	0.53	0.47	0.16	0.14	0.75	0.70	0.56	0.52	1.0	1.0	50	45	6.82	6.3	364	339
P3C	0.08	0.05	0.17	0.16	0.12	0.08	0.69	0.43	0.13	0.12	0.98	0.95	3	2	7	4.7	1610	625
PROCLUS	0.49	0.49	0.44	0.44	0.11	0.11	0.53	0.53	0.65	0.65	0.67	0.67	64	64	8	8	766	766
SCHISM	0.37	0.23	0.62	0.52	0.05	0.01	0.43	0.11	0.29	0.21	1.0	0.93	494	121	4.3	2.8	23031	391
STATPC	0.22	0.22	0.56	0.56	0.06	0.06	0.12	0.12	0.14	0.14	1.0	1.0	39	39	10	10	18485	16671
SUBCLU	0.24	0.18	0.58	0.38	0.04	0.01	0.39	0.04	0.30	0.13	1.0	1.0	10881	709	3.6	2	26047	2250



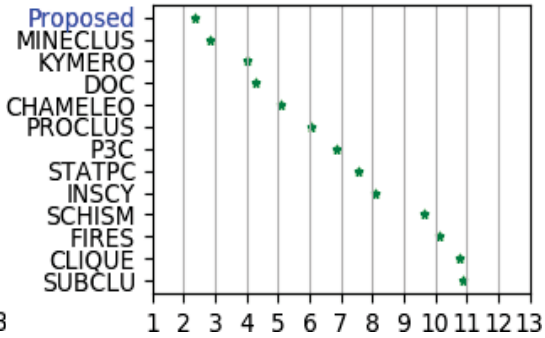
(a) Ranks based on F-Measure



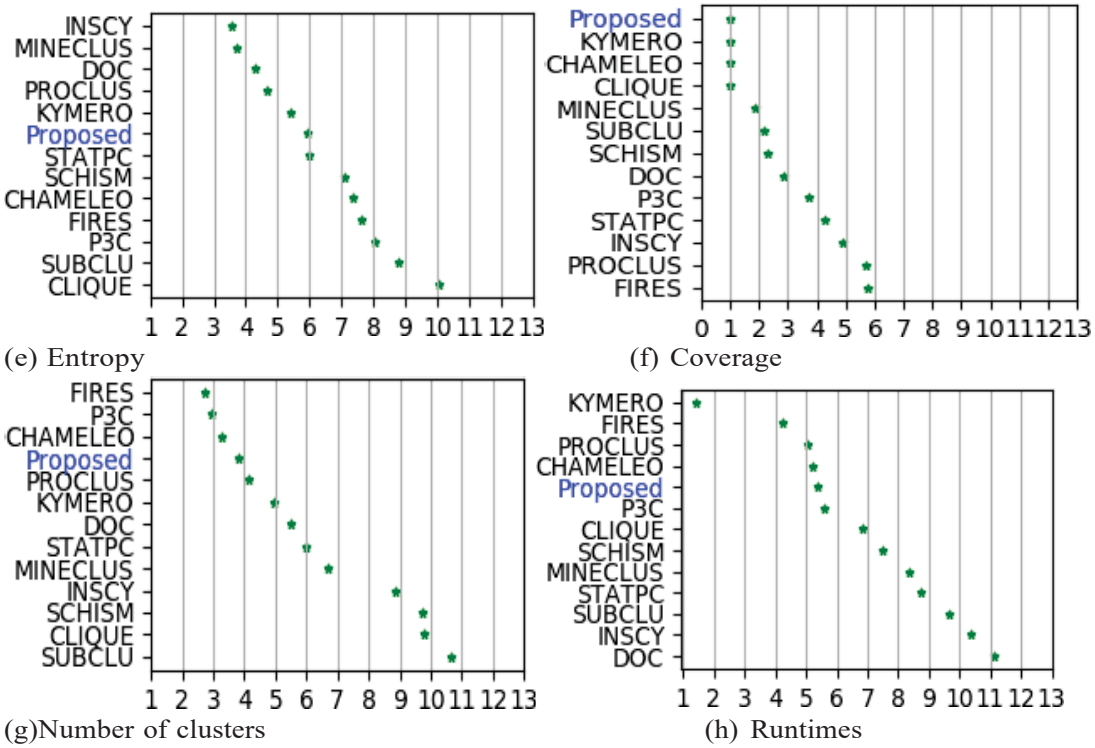
(b)Ranks based on Accuracy



(c) CE



(d) RNIA



(e) Entropy (f) Coverage

(g)Number of clusters (h) Runtimes

Figure 1: Rankings of different clustering algorithms

In contrast, algorithms like INSCY, MINECLUS, and others excel when it comes to the entropy metric. However, they struggle to effectively control the quantity of generated clusters, as depicted in Figure 1(g). These algorithms often generate a significant number of clusters, resulting in a lower ranking compared to the proposed approach. However, when considering F-Measure (Figure 1(a)), the proposed method may not claim the top spot but still positions itself among the top-performing approaches.

(b) Synthetic Data Sets

A similar procedure is followed by solutions for 20 datasets to that of the real-life. Each dataset undergoes 10 runs, and the recorded highest and least scores are outlined in Table 2. The proposed technique draws inspiration from the ChameleoClust process, our emphasis lies in directly comparing it with ChameleoClust across various synthetic datasets, streamlining the analysis. Figure 2 visually presents the best metric values attained by both approaches for these 20 datasets, along with their averages. These graphs specifically concentrate on key metrics, with the metric score versus the average number of clusters obtained after 10 runs for each dataset (Fig. 2(a) to 2(e)). From these graphs, we can make the following observations:

- The proposed algorithm generates several clusters, typically between 10 and 15, closely aligning with the quantity of clusters for datasets, which is 20. In contrast, ChameleoClust produces a broader range of clusters, spanning from 9 to 16. P3C and KymereoClust are other approaches which create clusters within the ranges of 6-16 and 9-14, respectively. The remaining algorithms produce even larger numbers of clusters and are not included in the comparison.
- In terms of the main evaluation metrics, the proposed algorithm stands out in FM, CE, and RNIA. Yet, ChameleoClust performs well in the accuracy metric. ChameleoClust also excels in terms of entropy, possibly because it generates a larger number of clusters assessed to the presented technique.

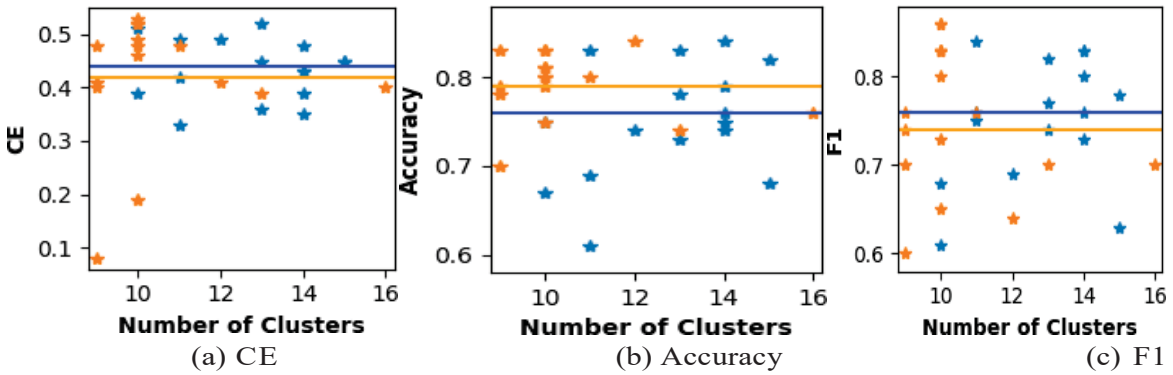
(c) Results on Categorical and Big Data Sets

For evaluating the effectiveness while applying various objectives, conducted experiments on 2 definite and 3 large datasets. We implemented and tested single and multi-objective optimization, which focuses merely on the ICC of the proposed process. The datasets utilized were gathered from UCI [183]. Additionally, two

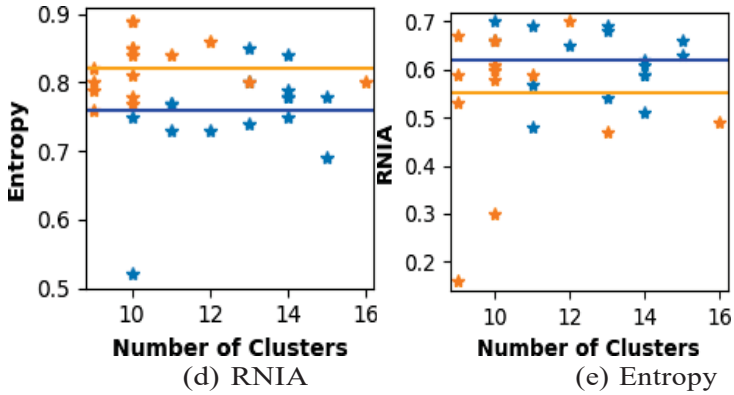
categorical datasets, Soybean (with 50 occurrences and 40 size) and the Molecular dataset (with 1640 occurrences and 72 size), were also considered.

Table 2: For the synthetic data sets

Dataset	F-Measure		Accuracy		CE		RNIA		Entropy		NumCluster		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
D05	0.61	0.35	0.75	0.52	0.51	0.33	0.7	0.54	0.75	0.55	12	10	2.78	1.86	2563	2476
D10	0.75	0.57	0.69	0.61	0.42	0.25	0.57	0.49	0.77	0.68	15	11	5.78	4.88	4631	4200
D15	0.69	0.52	0.74	0.59	0.49	0.34	0.65	0.56	0.73	0.67	16	12	10.89	6.76	5351	5298
D20	0.73	0.54	0.74	0.64	0.43	0.27	0.59	0.48	0.75	0.66	17	14	12.65	8.94	6949	6557
D25	0.74	0.57	0.78	0.62	0.52	0.39	0.68	0.55	0.8	0.73	16	13	16.84	10.88	7058	6969
D50	0.63	0.39	0.68	0.54	0.45	0.27	0.66	0.48	0.69	0.47	17	15	19.45	13.7	15181	13017
D75	0.68	0.23	0.67	0.49	0.39	0.25	0.66	0.44	0.52	0.38	17	10	35.3	9	24315	24315
N10	0.84	0.70	0.83	0.75	0.49	0.28	0.69	0.49	0.77	0.70	17	11	13.79	7.3	6428	6122
N30	0.77	0.62	0.73	0.64	0.36	0.26	0.54	0.39	0.74	0.64	16	13	13.2	8.4	7491	7436
N50	0.8	0.66	0.79	0.63	0.35	0.25	0.51	0.37	0.78	0.69	17	14	12.2	8.7	7333	7123
N70	0.76	0.46	0.61	0.49	0.33	0.22	0.48	0.28	0.73	0.58	16	11	13.15	6.1	7499	7119
S1500	0.82	0.69	0.83	0.78	0.45	0.35	0.69	0.56	0.85	0.74	16	13	13.78	10.28	6097	5869
S2500	0.83	0.68	0.84	0.74	0.48	0.33	0.59	0.49	0.84	0.72	16	14	12.45	9.85	7123	7110
S3500	0.76	0.65	0.75	0.71	0.43	0.33	0.62	0.53	0.79	0.7	15	14	14.65	11.11	7421	6733
S4500	0.83	0.66	0.76	0.69	0.39	0.27	0.61	0.53	0.78	0.67	17	14	12.85	10.5	7372	7341
S5500	0.78	0.65	0.82	0.71	0.45	0.31	0.63	0.52	0.78	0.71	16	15	13.24	8.76	6955	6927



(a) CE (b) Accuracy (c) F1



(d) RNIA (e) Entropy

Figure 2: Comparison of Chameleo Clust (orange) and proposed (blue) method

Table 3: categorical and big data sets concerning valuation metrics

Dataset	F-Measure		Accuracy		CE		RNIA		Entropy	
	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO
HTRU	0.72	0.81	0.95	0.97	0.11	0.33	0.58	0.39	0.74	0.77
Crowd Source	0.24	0.27	0.59	0.71	0.23	0.17	0.73	0.77	0.41	0.52
Magic	0.61	0.72	0.70	0.74	0.16	0.20	0.42	0.66	0.15	0.23
Soyabin	0.62	0.80	0.57	0.79	0.19	0.37	0.32	0.63	0.40	0.63
Molecular	0.53	0.66	0.58	0.70	0.17	0.33	0.78	0.94	0.11	0.24

The resemblance in 2 data points is determined by comparing respective characteristics. Table 3 shows that utilizing multiple objectives (MOO) yields better outcomes than single-objective approaches. MOO allows greater flexibility in selecting a solution.

(d) Comparison with Existing Overlapping Methods

Our proposed method is subjected to a comparative analysis against several existing overlapping techniques, namely OKM, OKMED, and WOKM [4]. This comparison is performed on various datasets, including Iris [4], Yeast [52], Scene [4], and Emotions [4]. Importantly, these existing methods necessitate the user to specify the number of clusters beforehand, while our proposed method autonomously determines the number of clusters.

For the Yeast, Scene, and Emotions datasets, which are multi-label datasets, the existent amount of clusters matches the sum of labels. To ensure a fair comparison, we modified our proposed method to generate a few clusters that match the entire labels. We assess the results using precision, recall, and F-measure metrics, and the findings are shown in Table 4. Values in bold indicate the best results obtained for a specific dataset in relation to a particular metric. The results in Table 4 clearly show the implementation of proposed method is better than other existing techniques.

Table 4: Results on real data sets considering the metrics

	Precision				**Recall**				**F-Measure**			
Iris	OKM	OKMED	WOKM	Proposed	OKM	OKMED	WOKM	Proposed	OKM	OKMED	WOKM	Proposed
Emotions	0.57	0.61	0.62	0.9176	0.98	0.88	0.98	0.9210	0.72	0.71	0.76	0.9193
Scene	0.49	0.49	0.49	0.8351	0.65	0.53	0.65	0.9759	0.56	0.50	0.56	0.8787
Yeast	0.23	0.24	0.21	0.3623	0.94	0.74	0.59	0.4419	0.36	0.36	0.31	0.3981

Furthermore, our explained method is subjected to a comparative assessment against other existing algorithms, specifically OKM and KHM-OKM (K-harmonic means overlapping K-means) [5]. In order to ensure an equitable evaluation, we employed the same real-world datasets that were utilized for testing the existing methods, including Breast cancer Wisconsin (Original) [5], Indian liver patients [5], Iris, Heart disease (Statlog) [5], Lung Cancer [5], and more. The evaluation is conducted using precision, recall, and F-measure metrics, with the best-performing results highlighted in bold. The findings tabulated in Table 4.5 clearly indicate that our method excels in generating high-quality clusters, as it consistently outperforms other algorithms in most cases.

Table 5: real-life data sets

Datasets	Precision			Recall			F-Measure		
	OKM	KHM-OKM	Proposed	OKM	KHM-OKM	Proposed	OKM	KHM-OKM	Proposed
Breast cancer Wisconsin (Original)	0.8471	0.8471	0.9286	0.9846	0.9846	0.9677	0.9107	0.9107	0.9478
Indian liver patients	0.5927	0.5927	0.6449	0.9229	0.9176	0.9998	0.7218	0.7180	0.784
Iris	0.57	-	0.9176	0.98	-	0.9210	0.72	-	0.9193
Heart disease (Statlog)	0.4957	0.4957	0.7626	0.7733	0.7733	0.7624	0.6041	0.6041	0.7625
Lung Cancer	0.4315	0.4645	0.7564	0.7365	0.6563	0.7194	0.5441	0.5439	0.7235

Furthermore, our proposed method is subjected to a comparative assessment against other existing algorithms, specifically OKM and KHM-OKM (K-harmonic means overlapping K-means) [5]. To ensure an equitable evaluation, we utilized the same real-world datasets that were employed for testing the existing methods, including Breast Cancer Wisconsin (Original) [5], Indian Liver Patients [5], Iris, Heart Disease (Statlog) [5], Lung Cancer [5], and more. The evaluation is conducted using precision, recall, and F-measure metrics, with the best-performing results highlighted in bold. The findings presented in Table 5 conclusively demonstrate that our proposed method consistently generates high-quality clusters, as it outperforms other algorithms in most cases.

(e) Proposed Method in Bi clustering with real life application

To show the practicality of our developed method, we selected two datasets: Human Large B Cell Lymphoma [6] and Yeast [6].We rigorously tested our proposed method using a variety of evaluation metrics as detailed in Section. The comparative outcomes shown in Table 6 (for the human dataset) and Table 7 (for the yeast dataset). Notably, our proposed approach consistently outperforms existing methods in most cases, as indicated by the highlighted values.

Table 6: Results for Human Large B Cell Lymphoma data set

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best (Max)	Average	Best(min)
Proposed	285.63	12.21	1569.86	6223.36	0.162	0.127
SGAB	855.36	572.54	2222.19	5301.83	0.5208	0.3564
MOPSOB	927.47	745.25	4348.2	8745.65	0.213	0.09
MOGAB	801.37	569.23	2378.25	5377.51	0.4710	0.2283
OPSM	1249.73	43.07	6374.64	11854.63	0.2520	0.1024
RWB	1185.69	992.76	1698.99	3575.40	0.7227	0.3386
BiVisu	1680.23	1553.43	1913.24	2468.63	0.8814	0.6537
ISA	2006.83	245.28	4780.65	14682.47	0.6300	0.0853
Bimax	387.71	96.98	670.83	3204.35	0.7120	0.1402

Table 7: Results of the proposed method on Yeast data set obtained by different algorithms.

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best (Max)	Average	Best(min)
Proposed	58.78	10.91	2419.61	5738.32	0.07	0.032
SGAB	198.88	138.75	638.23	3605.93	0.4026	0.2714
MOPSOB	218.54	200.58	789.85	1254.56	0.28	0.16
MOGAB	185.85	116.34	932.04	3823.46	0.3329	0.2123
OPSM	320.39	118.53	1083.24	3804.56	0.3962	0.0012
RWB	295.81	231.28	528.97	1044.37	0.5869	0.2788
BiVisu	290.59	240.96	390.73	775.41	0.7770	0.3940
ISA	281.59	125.76	409.29	1252.34	0.7812	0.1235
Bimax	32.16	5.73	39.53	80.42	0.4600	0.2104

VII. Chapter Summary

An overlapping subspace clustering approach based on multi-objective optimization is explained in this chapter. We have devised a novel objective function, referred to as the MNR-index, to optimize the inclusion of overlapping objects. Moreover, we improved the existing mutation operators to enhance our capability to discover effectively and efficiently. We conducted experiments on numerous real and synthetic datasets and compared our results with those from various established methods. This comparative analysis underscores the superior performance of our method in most cases.

Additionally, we demonstrate the application of subspace clustering in bi-clustering gene expression profile datasets. This chapter enables the clustering of objects with overlapping attributes, creating overlapping subspace clusters within the framework of multi-objective optimization. Our analysis of the results obtained in both this chapter and the previous one indicates that subspace clustering approaches are valuable for handling high-dimensional data. However, in today's context, many application domains continuously generate features in an online fashion. In such scenarios, not all features may be available initially and may arrive continuously. To address the continuous arrival of features, the following chapter introduces an approach for feature selection aimed at selecting the optimals.

References

[1] B. L. Welch, "The generalization ofstudent's' problem when several different population variances are in- volved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.

[2] M. Hund, I. Färber, M. Behrisch, A. Tatu, T. Schreck, D. A. Keim, and T. Seidl, "Visual quality assessmentof subspace clusterings," in *Workshop on Interactive Data Exploration and Analytics (IDEA'16)*, 2016, pp. 53–62.

[3] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[4] G. Cleuziou, "Two variants of the okm for overlapping clustering," in *Advances in Knowledge Discovery and Management*. Springer, 2010, pp. 149–166.

[5] S. Khanmohammadi, N. Adibeig, and S. Shanehbandy, "An improved overlapping k-means clustering methodfor medical applications," *Expert Systems with Applications*, vol. 67, pp. 12–18, 2017.

[6] U. Maulik, A. Mukhopadhyay, and S. Bandyopadhyay, "Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm," *IEEE Transactions on information technology in Biomedicine*, vol. 13, no. 6, pp. 969–975, 2009.

[7] J. Liu, Z. Li, F. Liu, and Y. Chen, "Multi-objective particle swarm optimization biclustering of microarray data," in *Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE*

International Conference on. IEEE, 2008, pp. 363–366.

[8] F. Angiulli and C. Pizzuti, "Gene expression biclustering using random walk strategies," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2005, pp. 509–519.

[9] B. L. Welch, "The generalization ofstudent's' problem when several different population variances are in- volved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.

[10] A. Ghosh, B. C. Dhara, and R. K. De, "Selection of genes mediating certain cancers, using a neuro-fuzzy approach," *Neurocomputing*, vol. 133, pp. 122–140, 2014.