Multi-Objective Multi-View Clustering Ensemble based on Evolutionary Approach

Abdul Wahid
Victoria University of Wellington
New Zealand
abdul.wahid@ecs.vuw.ac.nz

Xiaoying Gao
Victoria University of Wellington
New Zealand
xgao@ecs.vuw.ac.nz

Peter Andreae
Victoria University of Wellington
New Zealand
pondy@ecs.vuw.ac.nz

***Abstract*—Clustering ensembles is a clustering technique which derives a better clustering solution from a set of candidate clustering solutions. Clustering ensemble methods have to address two distinct but interlinked problems: *Generating multiple candidate solutions* from the data and *producing a final clustering solution*.**

Our recently proposed clustering ensembles method (MMOEA) based on NSGA-II used multiple views to address the first problem and a novel cluster oriented approach to address the second problem. MMOEA used a simple crossover method to explore the search space and three objective functions to determine the quality of a candidate clustering solution. The use of a simple crossover method led to slow convergence and using three objectives in NSGA-II framework is often discouraged.

This paper presents a new clustering ensemble method, which introduces new ideas for crossover, mutation, tuning steps and two objective functions (instead of three) in an evolutionary process. The results show that our new method outperforms recent methods for clustering ensembles on different multi-view datasets.

***Index Terms*—Clustering Ensemble, Multi-Objective Optimization, Evolutionary Algorithm**

I. Introduction

Clustering is a widely applied technique for analyzing data by organizing objects into different groups. A key challenge in clustering is that it is possible to have more than one good solution. In the last 50 years, thousands of different clustering methods have been proposed. In recent years, researchers have focused on getting better results by utilizing the fusion of different clustering methods. Clustering methods that try to combine different clustering solutions are commonly referred as Clustering Ensembles [1].

Clustering ensemble methods use a two step clustering process: step 1 generates the candidate clustering solutions, and step 2 constructs a single candidate clustering solution from the generated candidate clustering solutions.

There are different variations for clustering ensembles, but median partition based clustering ensembles are the best approach so far for step 2 of clustering ensembles [2]. The median partition approach forms a final solution by selecting a single candidate clustering solution from a set of candidate clustering solutions. A common way of selecting the final clustering solution is to pick the candidate clustering solution that has a maximum average similarity to all generated candidate clustering solutions.

Since different clustering solutions generally consist of high and low quality[1] clusters, a selected final clustering solution is also limited to have a mixture of high and low quality clusters. We propose a better clustering solution by applying the cluster oriented approach which combines high quality clusters from different clustering solutions.

Step 2 of clustering ensembles is heavily dependent upon step 1. If the generated candidate clustering solutions are not diverse and do not include a wide range of clusters then the final clustering solution might not be able to produce better results. One approach is to use multiple views for generating diverse clusters in step 1.

Recently, evolutionary approaches have become popular for multi-objective clustering ensembles because of their ability to provide good results [3]. Our recent work MMOEA [4] uses multiple views to generate an initial set of candidate clustering solutions and then applies an evolutionary approach (NSGA-II) using a simple crossover method and three objective functions to select better quality clusters. However, the simple crossover method might not result in diverse clustering solutions and hence needs to be replaced by better crossover functions. Moreover, mutation and tuning steps can help in exploring interesting search space and faster convergence.

Köppen et.al, argued that NSGA-II is not suitable for solving many (more than two) objective optimization problems [5]. Therefore the use of three objective functions is questionable and it would be better to either use NSGA-III or reduce the number of objectives.

This paper introduces a new approach for clustering ensembles i.e. Multi-Objective Multi-View Ensemble Clustering (*MOMVEC*) based on MMOEA and has the following innovations.

1) Developing crossover methods for generating new clusters for candidate clustering solutions.
2) Developing mutation methods for splitting and merging clusters.
3) Developing multi-objective fitness function for multi-objective optimization problem.

[1] Generally, the high quality clusters have low intra-cluster distances and high inter-cluster distance from other clusters

978-1-4799-7492-4/15/$31.00 © 2015 IEEE

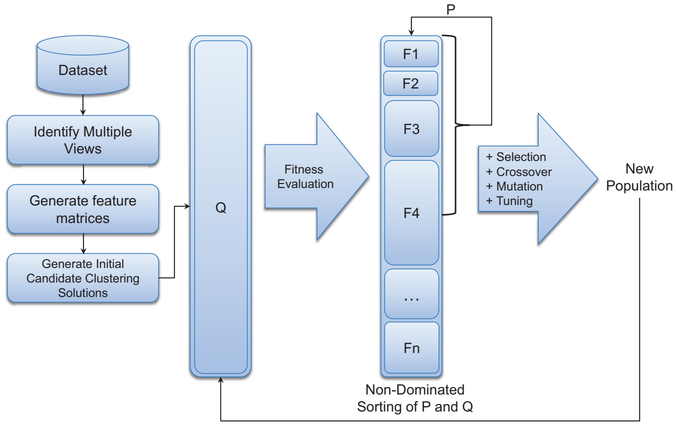


Fig. 1: Overview of MOMVEC clustering method

Section II discusses related work; section III explains the proposed method; section IV describes the experimental setup; section V provides the experiment results with discussions and lastly section VI concludes the paper.

II. Related Work

The majority of the clustering ensemble methods formulate the clustering as a single objective optimization problem, but some use multiple objectives. Handl and Knowles proposed a multi-objective evolutionary approach for clustering ensembles known as multi-objective clustering algorithm with k-determination (MOCK) based on PEAS-II [3], [6]. Their fitness function was based on two objectives: connectedness and compactness of the cluster. These two objectives were derived from link based clustering methods and the k-means algorithm.

Bandyopadhyay et al. [7] proposed a multi-objective evolutionary algorithm based on fuzzy clustering. Their proposed fitness function used two objective functions: J_m criterion [8] and Xie-Beni index [9].

Korkmaz et al. [10] proposed two objectives and used an encoding scheme based on linkage to reduce the redundancy of the set of candidate clustering solutions. The first objective was to minimize the number of clusters and the other was to minimize the intra-cluster variance. This approach used Pareto dominance to find a diverse set of nondominated clustering solutions.

Other prominent approaches for multi-objective clustering ensembles algorithms include [11], [12], [13]. Generally, multi-objective clustering ensembles uses two objectives and their objective criteria mainly focus on minimizing the intra-cluster distance. A detail analysis of multi-objective approaches can be found in [14], [15].

Clustering ensemble methods are currently limited to use only a single view of the data. Some clustering methods use multiple views in the clustering process, but are not categorized as clustering ensembles e.g. [16], [17] etc. Our proposed work is more related to MMOEA [4], a clustering ensemble method that uses multiple views.

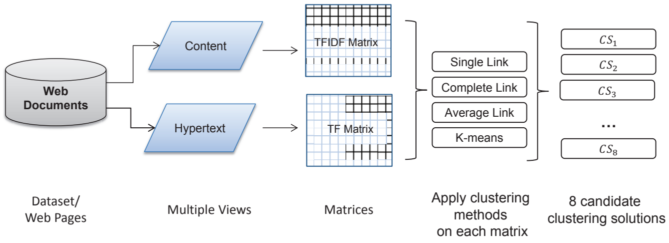


Fig. 2: Process of generating 8 different candidate clustering solutions.

III. The Method

Our Multi-objective Multi-view Ensemble Clustering (MOMVEC) is based on NSGA-II [18], which generates multiple candidate clustering solutions from multiple views of the data and then forms a better clustering solution by using a multi-objective evolutionary approach for selecting a set of high quality clusters from the candidate clustering solutions.

Figure 1 presents an overview of MOMVEC. The first step of the MOMVEC method is to identify multiple views of the dataset. Then these views are used to generate multiple feature matrices to which clustering algorithms are applied to generate multiple candidate clustering solutions. These initial candidate solutions are then encoded (genetic representation) as an initial population and their fitness is evaluated on the basis of two criteria.

The evolutionary process is then applied. The current population consists of Q (the set of new individuals just created) and P (the elite individuals obtained from the previous iteration).

After fitness evaluation on Q, we use the NSGA-II to compute Pareto fronts (F1, F2 etc) and rank all the candidate clustering solutions (i.e. P and Q). If the stopping criteria is not met then we retain the top half of the ranked candidate solutions in P and Q and then perform *Selection* then *Crossover*, *Mutation* followed by *Tuning* steps to generate a set of new candidate clustering solutions (Q).

The following sections describe the method in more details.

A. Generating Initial Candidate Clustering Solutions

Clustering ensembles commonly generate an initial set of candidate clustering solutions by applying different types of clustering algorithms on one feature matrix. Instead, we use multiple matrices from multiple views of the data.

Figure 2 shows the process of generating initial candidate clustering solutions from the WebKB2 dataset. The content and hypertext of the inlinks of the web pages are the two views of the web pages[1]. The feature matrices are generated by using the standard Term Frequency Inverse Document Frequency (TFIDF) and Term Frequency (TF) weighting schemes on the content (all terms) and hypertext of the web pages. Four popular clustering algorithms - single link, complete link, average link and k-means - are then applied to the two feature matrices to generate eight candidate clustering solutions.

[1] These two views were predefined and provided with the dataset.

B. Genetic Representation of Candidate Clustering Solutions

Since the evolutionary approach requires a genetic representation for candidate clustering solutions, we used a matrix based binary encoding scheme [15] in which a clustering solution or individual is represented as a $k \times N$ matrix. The k rows represent clusters and the N columns represent documents.

Figure 3 depicts a sample clustering solution and its matrix based encoding is described in Figure 4. The value 1 in a cell of the matrix means that the document is assigned to the cluster. The key advantage of this representation is that it can represent overlapping clusters. Note that this representation would allow a clustering in which some documents are not allocated to any cluster.

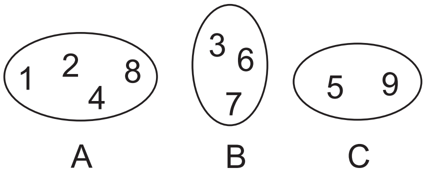


Fig. 3: Sample Clustering.

	1	2	3	4	5	6	7	8	9
A	1	1	0	1	0	0	0	1	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1

Fig. 4: Matrix-based binary encoding scheme of an individual where columns represent the document ids and rows represent clusters.

C. Fitness Evaluation

Fitness evaluation of candidate clustering solutions is based on the following two objectives criteria.

1) Intra-cluster distance: Minimize the distances between the objects within each cluster of a clustering solution.
2) Inter-cluster similarity: Minimize the similarity between each pair of clusters in a clustering solution.

Both objectives favor the small clusters, therefore we added a weighting parameter based on cluster size to the first objective which favors the big clusters. The two objectives trade off against each other. One gives more importance to clustering solutions that have big clusters of closely related objects and the other gives preference to small clusters that are different from each other. The first objective function is defined as follows:

$$\Phi_a(C) = \frac{1}{|C|} \sum_{c \in C} \frac{1}{\sqrt{|c|}} \left(\frac{1}{|c|(|c|-1)} \sum_{d,d' \in c} \delta(d,d') \right)$$

where c is a cluster in clustering solution C, d and d' are documents in cluster c, δ is a distance function which computes the distance between two documents. $\Phi_a(C)$ is the average over all clusters of the cluster spread, weighted by a function of cluster size. The cluster spread is the average distance between pairs of documents in the cluster. We used cosine function for computing the similarity/distances between the documents. The cosine function provides a value from 0-1, where a higher value indicates that documents are similar. Cosine similarity function can also be used as distance function if the value of cosine similarity function is subtracted from 1. $\delta(d,d')$ represents the 1-cosine whereas $\gamma(d,d')$ represents cosine of d, d'. The second objective function is defined as follows:

$$\Phi_e(C) = \frac{1}{|C|(|C|-1)} \sum_{c,c' \in C \wedge c \neq c'} \left(\frac{1}{|c||c'|} \sum_{d \in c, d' \in c'} \gamma(d,d') \right)$$

where c is a cluster in clustering solution C, d and d' are the document in cluster c, γ is a similarity function which computes the similarity between two documents. $\Phi_e(C)$ is the average over all pairs of clusters of the similarity of the pair of clusters. The similarity of two clusters is the average similarity of the pairs of documents from the two clusters.

Both objective functions are coded in such a way that their values are required to be minimized. Therefore our algorithm must solve the multi-objective optimization problem which attempts to find a solution that minimize both $\Phi_a(C)$ and $\Phi_e(C)$.

The evolutionary approach not only requires a good fitness evaluation criteria but also needs a mechanism to generate new and diverse candidate clustering solutions in order to avoid local optima.

D. Selection

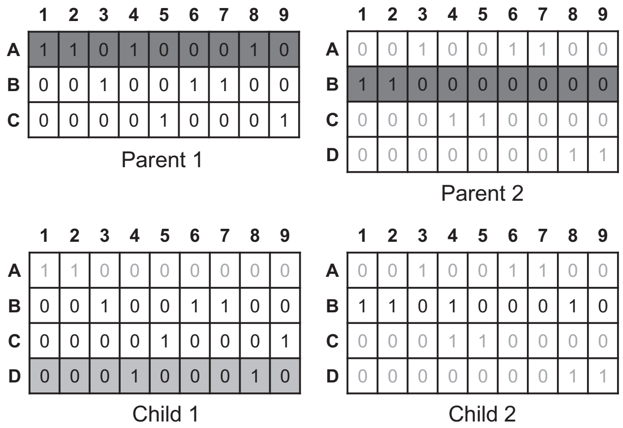
The selection step selects a number of candidate solutions from the previous iteration to generate a set of new candidate clustering solutions, based on the NSGA-II selection method which uses ranks and crowding distance of the candidate clustering solutions [18].

E. Crossover

In evolutionary approaches the term crossover indicates a process for generating a new individual from two previous individuals. This research work used two crossover methods: the row-wise and the column-wise crossover method.

The row-wise method swaps a randomly chosen row from one parent with a randomly chosen row from the other parent, creating two new children. If either child has any columns that are all zeros (i.e. unassigned documents), it adds an extra row to that child containing these documents. This crossover method may result in overlapping clusters but the coverage will always be 100%.

Figure 5 shows an example of the row-wise crossover method in which the rows with dark gray background are swaped between Parent 1 and Parent 2 and as a result Child 1 and Child 2 are created. The extra row highlighted in light gray in Child 1 contains the unassigned documents after the swapping.



Parent 1

	1	2	3	4	5	6	7	8	9
A	1	1	0	1	0	0	0	1	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1

Parent 2

	1	2	3	4	5	6	7	8	9
A	0	0	1	0	0	1	1	0	0
B	1	1	0	0	0	0	0	0	0
C	0	0	0	1	1	0	0	0	0
D	0	0	0	0	0	0	0	1	1

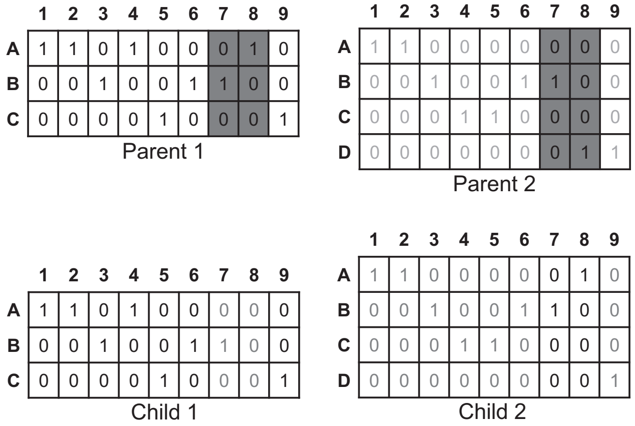
Child 1

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	0	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1
D	0	0	0	1	0	0	0	1	0

Child 2

	1	2	3	4	5	6	7	8	9
A	0	0	1	0	0	1	1	0	0
B	1	1	0	1	0	0	0	1	0
C	0	0	0	1	1	0	0	0	0
D	0	0	0	0	0	0	0	1	1

Fig. 5: Example of row-wise crossover method.



Parent 1

	1	2	3	4	5	6	7	8	9
A	1	1	0	1	0	0	0	1	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1

Parent 2

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	0	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	1	1	0	0	0	0
D	0	0	0	0	0	0	0	1	1

Child 1

	1	2	3	4	5	6	7	8	9
A	1	1	0	1	0	0	0	0	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1

Child 2

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	1	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	1	1	0	0	0	0
D	0	0	0	0	0	0	0	0	1

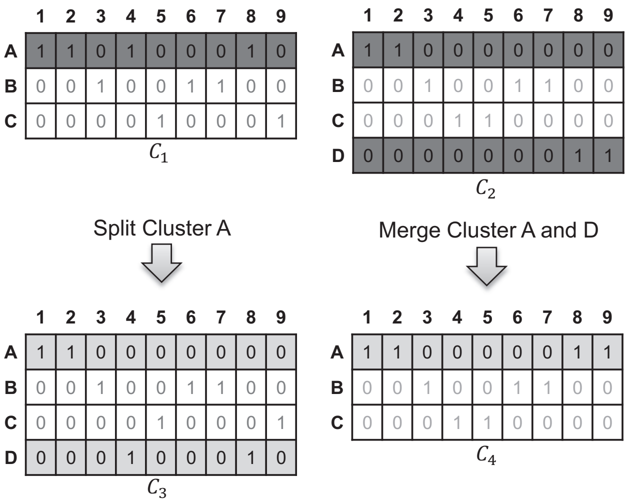
Fig. 6: Example of column-wise crossover method.

The column-wise method swaps randomly (from 2 to 10) chosen columns of one parent with the corresponding columns of the other parent. If the parents have different number of clusters, then excess rows are given the zero value for chosen columns. Additional step in the column-wise crossover is to remove the empty cluster. This type of crossover does not affect the overlapping of documents but may result in less than 100% coverage.

Figure 6 shows an example of column-wise crossover method in which the columns with dark gray background are swaped between Parent 1 and Parent 2 and as a result Child 1 and Child 2 are created. The the column 8 in Child 1 contains zero value for all rows indicating that the document was not assigned to any cluster.

F. Mutation

In this research work, two different types of mutation methods are used. Split-mutation splits big clusters into two clusters and merge-mutation merges two small clusters into one cluster. We used a random approach for split-mutation which forms two clusters randomly from one cluster. The merge-mutation is based on inter-cluster distances and merges two small clusters in a candidate clustering solution that have minimum inter cluster distance.



C_1

	1	2	3	4	5	6	7	8	9
A	1	1	0	1	0	0	0	1	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1

C_2

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	0	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	1	1	0	0	0	0
D	0	0	0	0	0	0	0	1	1

C_3

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	0	0
B	0	0	1	0	0	1	1	0	0
C	0	0	0	0	1	0	0	0	1
D	0	0	0	1	0	0	0	1	0

C_4

	1	2	3	4	5	6	7	8	9
A	1	1	0	0	0	0	0	1	1
B	0	0	1	0	0	1	1	0	0
C	0	0	0	1	1	0	0	0	0

Fig. 7: Example of split and merge mutation method.

Figure 7 shows two examples of mutation. On the left, split-mutation is applied to C_1 to generate C_3. Cluster (row) A is randomly selected and split into two clusters A and D. On the right, merge-mutation is applied to C_2. Cluster A and D in C_2 are merged into cluster A in C_4.

G. Tuning

The newly generated candidate clustering solutions are mostly based on a random approach and may not represent sensible clusterings. This would cause the evolutionary approach to converge slowly. Therefore we need a local search mechanism to find local optima more quickly. This tuning is based on k-means and includes the following two steps for all newly generated candidate clustering solutions.

1) Calculating cluster centroids.
2) Relocating each document to the cluster with the nearest centroid.

These two steps are repeated until there is no change to centroids and no more relocation of the documents.

H. Algorithm: MOMVEC

The algorithm MOMVEC developed in this research work uses a multiple objectives approach for selecting the final clustering solution. The algorithm ranks individuals iteratively based on Pareto ranking and crowding distance like NSGA-II. It has the following properties:

- it finds *Pareto non-dominated* candidate solutions in a multi-objective optimization and uses *Pareto ranking* to sort all candidate solutions.
- it uses the concept of elitism, which means keeping the best candidate solutions.
- it preserves the diversity in a set of candidate clustering solutions, using crossover and mutation methods.

Let $\mathcal{E}$ be a multi-objective optimization problem of the form: $p^* = \text{argmin}_p \{f_1(p), ...f_n(p)\}$. Then *Pareto dominance*, *Pareto non-dominance* and *Pareto ranking* are defined as follows [19]:

Algorithm 1 MOMVEC

Input: initial candidate clustering solutions P_{init} and maximum number of iterations max.
Output: Final clustering solution $\mathcal{C}$.

```
1:  $\mathcal{P} \leftarrow initializePopulation(P_{init})$
2:  for $i \leftarrow 1, max$ do
3:      Compute Pareto ranking for $\mathcal{P}$ and sort $\mathcal{P}$
4:      $\mathcal{P}^* \leftarrow$ top half from $\mathcal{P}$
5:      $Q \leftarrow$ Null
6:      for $i \leftarrow 1, size(\mathcal{P})/2$ do
7:          $p_1, p_2 \leftarrow$ select two candidate clustering solutions from $\mathcal{P}$
8:          $rand \leftarrow$ generate a random number from 1 to 6.
9:          if $rand = 1$ then
10:             $c_1, c_2 \leftarrow$ rowCrossover($p_1, p_2$)
11:         else if $rand = 2$ then
12:             $c_1, c_2 \leftarrow$ columnCrossover($p_1, p_2$)
13:         else if $rand = 3$ or $rand = 4$ then
14:             $c_1, c_2 \leftarrow$ Apply merge-mutation if possible, otherwise apply split-mutation on $p_1$ and $p_2$
15:         else if $rand = 5$ or $rand = 6$ then
16:             $c_1, c_2 \leftarrow$ Apply split-mutation if possible, otherwise apply merge-mutation on $p_1$ and $p_2$
17:         end if
18:         $Q \leftarrow$ Apply tuning on $c_1$ and $c_2$
19:     end for
20:     $\mathcal{P} \leftarrow \mathcal{P}^* \cup Q$
21: end for
22: $\sigma \leftarrow$ Compute Pareto ranking for $\mathcal{P}$
23: $\mathcal{P}^* \leftarrow \{p' \in \mathcal{P} : \forall p' \in \mathcal{P}, \sigma(p') = 1\}$
24: Select $\mathcal{C}$ from $\mathcal{P}^*$
```

Definition 1. (Pareto dominance) Let p and p' be two candidate solutions of $\mathcal{E}$. p is said to have a Pareto dominance over p' ($p \prec p'$) if and only if p has a lower value than p' on at least one objective function and has a lower or equal value on the remaining objective functions.

Definition 2. (Pareto non-dominated set) Let $\mathcal{P}$ be a set of candidate solutions of $\mathcal{E}$. $\mathcal{P}^*_\mathcal{E} \subseteq \mathcal{P}$ is a Pareto non-dominated solution set of $\mathcal{E}$ w.r.t. $\mathcal{P}$ if and only if $\forall_{p \in \mathcal{P}, p^* \in \mathcal{P}^*_\mathcal{E}}\ p \not\prec p^*$

Definition 3. (Pareto ranking) Let $\mathcal{P}$ be a population of individuals for $\mathcal{E}$. The Pareto ranking function $\sigma : \mathcal{P} \rightarrow \mathbb{N}^+$ for $\mathcal{E}$ is defined as follows.

$\mathcal{P}_1$ is the non-dominated subset of $\mathcal{P}$. For $i > 1$, $\mathcal{P}_i$ is the non-dominated subset of $\mathcal{P} \backslash \bigcup_{0<j<i} \mathcal{P}_j$. The Pareto rank of a candidate clustering solution p is the index of the subset it belongs to: $\forall p \in \mathcal{P}_i, \sigma(p) = i$.

The Pareto ranking function as described in Definition 3 provides a score for each candidate clustering solution in a set of candidate clustering solution $\mathcal{P}$.

Algorithm 1 describes the salient operation of our algorithm MOMVEC. It takes two arguments as inputs, the initial candidate clustering solutions and the maximum number of iterations. The initial candidate clustering solutions are the set of candidate solutions $\mathcal{P}$ generated from multiple views (as described in section III-A). The Pareto ranking function σ is calculated for current set of candidate clustering solutions $\mathcal{P}$ according to Definition 3 using the objective functions $\Phi_a(C)$ and $\Phi_e(C)$.

The ranking function σ sorts the candidate clustering solutions $\mathcal{P}$. The loop on line 6 generates a new set of candidate clustering solutions by applying different methods on $\mathcal{P}$. The row-wise and column-wise crossover methods are chosen with a probability of $\frac{1}{6}$ whereas the merge-mutation and split-mutation methods are chosen with a probability of $\frac{1}{3}$. The new candidate clustering solutions go under the tuning step and then merged with previous top rank (non-dominated) set of candidate clustering solutions $\mathcal{P}^*$.

Once the main loop is completed, the set of Pareto optimal solution $\mathcal{P}^*$ is computed from $\mathcal{P}$. The final clustering solution $\mathcal{C}$ is finally selected based on longest crowding distance from rank 1 Pareto front as described by [18].

IV. Experimental Setup

MOMVEC uses the following parameter settings: maximum number of generations = 1000, crossover probability $\frac{1}{6}$, mutation probability $\frac{1}{3}$, population size = 20. We used a random number of clusters for k-means algorithm and fixed number of clusters (provided by user) for all other algorithms to generate initial candidate clustering solutions. MOMVEC was compared with three commonly used single objective clustering ensembles and two multi-objective clustering approaches. The final clustering solutions of all methods were compared using three different metrics: Clustering Accuracy (CA) [20], F1-measure (F1) [21] and Rand Index (RI) values [22], which measures the quality of clustering solution against the provided gold standard clustering solutions for a given dataset[1].

This research work was tested on four two-view datasets WebKB2[2], WebKB4[2], Citeseer[3] and Cora[4], and one three-view dataset consisting of AMBIENT[5], MORESQUE[6] and ODP-239[7] datasets. These datasets are widely used for testing clustering algorithms and provide multiple views.

Our method MOMVEC was compared with five methods: CTS-Ensemble, SRS-Ensemble, ASRS-Ensemble, MOCK and MMOEA. The first three methods are three basic single objective clustering ensemble methods based on link-based pairwise similarity matrices [23].

MOCK is a multi-objective evolutionary algorithm recently proposed by [3]. The code was provided by its author, we

[1] Clustering is an unsupervised learning in which the true clustering solution or gold standards (generated by humans) for a given dataset is not available during the clustering process. Therefore, we can not use the evaluation metrics as an objective functions during the clustering process.

[2] http://www.cs.cmu.edu/~webkb/

[3] http://www.cs.umd.edu/~sen/lbc-proj/data/citeseer.tgz

[4] http://www.cs.umd.edu/~sen/lbc-proj/data/cora.tgz

[5] downloaded from http://credo.fub.it/ambient/

[6] downloaded from http://lcl.uniroma1.it/moresque/

[7] downloaded from http://credo.fub.it/odp239/

Dataset	CTS-Ens		SRS-Ens		ASRS-Ens		MOCK		MMOEA			MOMVEC		
	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	cmb	v1	v1+v2	cmb
WebKB2-Cornell	83.54	83.95	83.54	83.59	83.95	83.95	84.34	84.43	91.12	91.12	91.12	**91.12**	**91.12**	**92.51**
WebKB2-Texas	86.61	87.04	86.22	87.04	86.22	87.94	87.01	87.39	90.12	90.21	90.42	**90.15**	**90.21**	**91.45**
WebKB2-Washington	79.31	78.09	71.09	71.09	71.09	71.09	80.12	81.23	84.91	86.43	86.72	**87.11**	**88.19**	**92.13**
WebKB2-Wisconsin	73.83	75.16	73.83	74.16	73.83	74.16	74.98	75.33	85.54	86.96	88.54	**88.18**	**90.18**	**92.88**
WebKB4-Cornell	56.41	57.95	56.92	57.69	56.41	57.69	58.01	59.98	68.12	68.82	70.12	**70.13**	**71.11**	**75.23**
WebKB4-Texas	71.12	59.89	71.12	59.36	72.36	60.17	72.36	60.17	73.01	74.87	75.45	**74.12**	**76.87**	**78.13**
WebKB4-Washington	68.26	68.83	69.57	69.65	69.13	69.65	69.82	70.23	72.17	72.51	74.17	**72.34**	**72.54**	**77.93**
WebKB4-Wisconsin	75.47	77.74	75.74	78.49	78.11	78.49	**79.91**	80.45	**79.91**	80.98	79.91	**79.91**	**81.54**	**83.91**
Citeseer	36.02	42.63	48.04	48.32	43.87	44.32	45.98	45.98	50.01	51.45	54.34	**51.12**	**53.32**	**60.12**
Cora	46.12	47.15	41.91	42.35	44.72	46.35	46.51	47.22	50.95	51.12	55.11	**51.31**	**52.32**	**60.15**

TABLE I: Clustering Accuracy computed on 10 different multi-view datasets.

Dataset	CTS-Ens		SRS-Ens		ASRS-Ens		MOCK		MMOEA			MOMVEC		
	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	cmb	v1	v1+v2	cmb
WebKB2-Cornell	91.28	91.28	91.28	91.28	91.28	91.28	91.54	**91.68**	**91.95**	**91.68**	**92.01**	**91.95**	**91.68**	**92.01**
WebKB2-Texas	90.14	92.14	**92.61**	92.61	**92.61**	92.61	92.56	**92.69**	**92.61**	**92.69**	**93.28**	**92.61**	**92.69**	**93.28**
WebKB2-Washington	88.45	88.71	88.71	88.71	88.71	88.71	89.91	89.91	90.71	91.74	92.14	**91.11**	**92.14**	**93.83**
WebKB2-Wisconsin	85.62	86.76	85.62	86.45	84.38	86.45	86.12	86.93	86.32	87.23	89.91	**87.12**	**89.12**	**92.79**
WebKB4-Cornell	66.17	67.21	66.52	67.56	66.52	67.56	67.56	68.32	69.12	70.34	71.12	**70.22**	**71.63**	**75.81**
WebKB4-Texas	65.09	47.97	65.09	47.97	65.09	47.97	66.01	48.13	67.19	67.73	70.01	**68.31**	**70.73**	**74.07**
WebKB4-Washington	50.44	63.67	55.43	63.48	56.24	63.67	58.23	64.39	62.99	67.16	71.25	**63.87**	**69.65**	**74.41**
WebKB4-Wisconsin	66.85	67.73	66.85	68.01	67.87	68.01	70.03	70.97	74.32	75.46	76.23	**75.15**	**76.75**	**78.14**
Citeseer	50.59	56.41	50.94	52.76	52.87	56.78	54.01	58.94	58.87	59.73	61.23	**61.18**	**61.73**	**68.29**
Cora	56.05	58.61	52.83	56.33	55.54	56.53	57.98	58.17	59.62	61.15	62.21	**60.54**	**62.17**	**63.54**

TABLE II: F1-measure computed on 10 different multi-view datasets

Dataset	CTS-Ens		SRS-Ens		ASRS-Ens		MOCK		MMOEA			MOMVEC		
	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	v1	v1+v2	cmb	v1	v1+v2	cmb
WebKB2-Cornell	72.38	72.94	71.29	72.94	71.94	72.94	71.01	72.94	72.66	72.94	75.21	**73.13**	**73.64**	**75.43**
WebKB2-Texas	76.72	76.91	76.14	76.33	76.14	76.14	76.99	76.34	77.89	78.31	79.89	**78.23**	**78.51**	**80.21**
WebKB2-Washington	67.04	68.41	56.86	58.41	57.01	58.41	68.19	69.12	70.41	75.12	79.32	**71.12**	**78.12**	**87.12**
WebKB2-Wisconsin	60.91	61.55	59.38	61.55	59.38	61.55	69.11	70.33	74.93	74.93	75.58	**75.45**	**78.93**	**88.32**
WebKB4-Cornell	61.02	61.54	61.92	61.96	61.02	61.96	63.12	63.76	70.79	75.12	76.17	**73.12**	**76.07**	**82.72**
WebKB4-Texas	64.19	48.13	66.73	48.13	66.73	48.13	67.53	50.11	72.65	73.12	74.65	**74.95**	**76.41**	**80.91**
WebKB4-Washington	66.73	67.12	66.17	67.96	65.52	67.96	72.12	73.01	73.13	74.76	76.21	**74.43**	**75.11**	**80.56**
WebKB4-Wisconsin	74.82	75.75	75.15	76.83	75.58	76.83	75.88	77.63	80.76	81.12	82.12	**82.94**	**83.45**	**85.14**
Citeseer	60.98	65.35	64.32	68.76	60.43	73.06	72.23	74.34	76.26	77.13	77.23	**78.26**	**79.36**	**85.21**
Cora	63.82	69.34	70.44	72.47	70.21	73.92	76.12	74.23	78.79	80.21	81.83	**80.11**	**81.32**	**89.75**

TABLE III: Rand Index computed on 10 different multi-view datasets

used standard parameters for generating the results on different views of the different datasets[1].

MMOEA is a multi-objective multi-view evolutionary algorithm recently proposed in [4]. We used standard parameters for the algorithm. This algorithm provides the facility to get the final results by combining the multiple views. MMOEA and MOMVEC both use the NSGA-II approach to compute the Pareto fronts and differs in terms of implementation of the algorithm, fitness function, crossover methods, mutation methods and tuning functions.

V. Results and Discussion

Tables I, II and III present the average values of 100 runs of the clustering methods in terms of CA, F1 and RI values respectively[2]. These values are computed on all datasets specified the in first column for their corresponding clustering methods. The v1 and v1+v2 columns for all clustering methods indicate that the value is computed for view one (terms) and a concatenation of two views (view one and view two) into a single feature matrix. The cmb column mentioned under MMOEA and MOMVEC indicates that the view one and view two both were used in the clustering methods separately having two different feature matrices.

The best values obtained in terms of CA, F1 and RI are shown in bold font for v1, v1+v2 and cmb. Overall, MOMVEC always performed at least equally well to the other clustering methods in terms of CA, F1 and RI and generally outperformed other clustering methods.

[1] http://personalpages.manchester.ac.uk/mbs/julia.handl/mock.html

[2] values are converted to percentages by multiplying them with 100 for better understanding

Dataset	CA Improvement			F1 Improvement			RI Improvement		
	v1	v1+v2	cmb	v1	v1+v2	cmb	v1	v1+v2	cmb
WebKB2-Cornell	0.000	0.000	1.525	0.000	0.000	0.000	0.647	0.960	0.293
WebKB2-Texas	0.033	0.000	1.139	0.000	0.000	0.000	0.437	0.255	0.401
WebKB2-Washington	2.591	2.036	6.238	0.441	0.436	1.834	1.008	3.994	9.834
WebKB2-Wisconsin	**3.086**	**3.703**	4.902	0.927	2.167	3.203	0.694	**5.338**	**16.856**
WebKB4-Cornell	2.951	3.328	7.288	1.591	1.834	6.594	**3.291**	1.265	8.599
WebKB4-Texas	1.520	2.671	3.552	1.667	**4.429**	5.799	3.166	4.499	8.386
WebKB4-Washington	0.236	0.041	5.069	1.397	3.708	4.435	1.778	0.468	5.708
WebKB4-Wisconsin	0.000	0.692	5.006	1.117	1.710	2.506	2.699	2.872	3.678
Citeseer	2.220	3.635	**10.637**	**3.924**	3.348	**11.530**	2.623	2.891	10.333
Cora	0.707	2.347	9.145	1.543	1.668	2.138	1.675	1.384	9.679

TABLE IV: Percentage improvement of MOMVEC compare to MMOEA

In table I, MOCK, MMOEA and MOMVEC have the maximum value **79.91** for view one on the WebKB4-Wisconsin dataset, however, MOMVEC outperformed all other clustering methods in v1+v2 and cmb views.

In table II, MMOEA and MOMVEC have the highest F1 values on WebKB2-Cornell dataset for all views. The SRS-Ens, ASRS-Ens, MMOEA and MOMVEC clustering methods have the highest value for view one, but MOMVEC outperformed other clustering methods when the views were combined on WebKB2-Texas dataset.

Table III shows that MOMVEC outperformed all other clustering methods in terms of RI on all views.

Table IV shows the percentage improvements of MOMVEC over MMOEA in terms of CA, F1 and RI values on datasets having two views. The highlighted values represent the maximum improvement achieved by the MOMVEC algorithm. In terms of CA, F1 and RI values the maximum improvement achieved by the MOMVEC is **10.637%** on Citeseer dataset, **11.530%** on Citeseer dataset and **16.856%** on WebKB2-Wisconsin dataset respectively. Overall MOMVEC shows a reasonable improvement over MMOEA in most of the cases.

We also computed CA, F1 and RI values on the combined dataset of 397 queries for all views[1] which also showed that MOMVEC outperformed other clustering methods. Figure 8 shows F1 values computed on combined dataset. The values for CA and RI also shows the same trend.

The results performed on view one and view two produced different clustering solutions. The CA, F1 and RI values which represent the quality of clustering were different for both views. Generally view one (terms) dominated view two (hypertext), therefore we only showed the results of view one. However, In some cases view two dominated the results. This variation of results on two views suggested that we should consider multiple views for clustering the data.

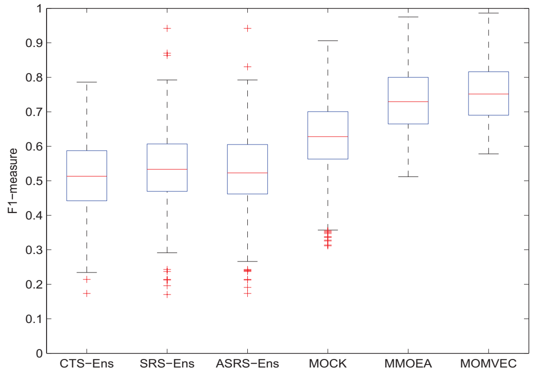


Fig. 8: box plot of F1-measure values on combined dataset

Simply concatenating the two views into a single feature matrix is not always a good solution. As shown in the results, the CA, F1 and RI values computed on WebKB4-Texas dataset for view v1+v2 was worse than v1 in case of CTS-Ensemble, SRS-Ensemble, ASRS-Ensemble and MOCK clustering methods. MOMVEC and MMOEA used multiple views separately therefore they got better results.

In order to get enough evidence to conclude that our proposed method MOMVEC provides a significant improvement, we performed the pairwise Wilcoxon statistical significance test [24] on CA, F1 and RI values computed on all datasets. The values for MOMVEC were treated as a control group and was compared individually with the values of the other clustering methods.

The p-values for two views and combined datasets calculated on Clustering Accuracy, F1-measure and RI values are less then **0.005**. The statistical test was performed for α = **0.05** and the results showed that the improvement of MOMVEC on all other clustering methods is statistically significant.

Figure 9 shows the performance of MOMVEC when we independently took out crossover, mutation and tuning steps. The y-axis represents the average Clustering Quality (Average CA) computed on all datasets and the x-axis represents the number of generations. The algorithm converged after 240 generations with all steps, 600 generations without the tuning step, 455 generations without the mutation step and 360 generations without the crossover step. This analysis provided the insight about the importance and impact of the each step. The tuning step (i.e. local search) had the greatest impact on the speed of convergence of the algorithm. The mutation methods were more important for speed of convergence (though not for

[1] we concatenated three views and formed on feature matrix for CTS-Ensemble, SRS-Ensemble, ASRS-Ensemble and MOCK and generated three feature matrices from views separately for MMOEA and MOMVEC for fair comparison

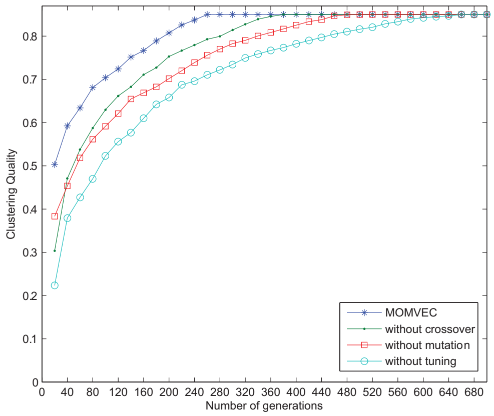


Fig. 9: Comparison of average clustering quality (CA) of MOMVEC, MOMVEC without tuning, MOMVEC without mutation, MOMVEC without crossover steps for 700 generations.

early accuracy) than the crossover methods.

VI. Conclusion

The main contribution of this work was a new clustering ensemble method (MOMVEC) based on MMOEA. MOMVEC introduced four new innovations: weighted intra-cluster distance in objective functions, row/column wise crossover methods, split/merge mutation methods and a tuning method in the multi-objective evolutionary process for refining the clustering solution.

MOMVEC outperformed other clustering ensemble methods. Even when restricted to a single view MOMVEC provided better results on the majority of the datasets and was never worse. The use of multiple views generated a diverse set of clusters and led to even better results. The results also lead to the conclusion that using a multi-objective approach is much better than using a single objective approach for clustering ensembles.

MOMVEC is able to produce both overlapping and non-overlapping clusters. The presented clustering approach automatically determines the number of clusters for final clustering solution. The limitation of this approach is that it requires multiple views to be predefined.

The current approach works well on small to medium size datasets having not more than 1500 features. Experiments indicate it is slow on large datasets with more then 4000 features. One of the remedies for this issue is to use dimensionality reduction techniques to reduce the number of features. However, using dimensionality reduction techniques may lead to loss of information.

Future directions for this work are to improve the scalability of MOMVEC on larger corpora by applying an effective dimensionality reduction technique and to automatically identify the multiple views of the data without using any predefined views or domain knowledge.

References

[1] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011.

[2] J.-P. Barthélemy and B. Leclerc, "The median procedure for partitions," *DIMACS series in discrete mathematics and theoretical computer science*, vol. 19, pp. 3–34, 1995.

[3] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 1, pp. 56–76, 2007.

[4] A. Wahid, X. Gao, and P. Andreae, "Multi-view clustering of web documents using multi-objective genetic algorithm," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 2625–2632.

[5] M. Köppen and K. Yoshida, "Substitute distance assignments in nsga-ii for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 727–741.

[6] J. Handl and J. Knowles, "Evidence accumulation in multiobjective data clustering," in *Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 543–557.

[7] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 5, pp. 1506–1511, 2007.

[8] L. I. Kuncheva and J. C. Bezdek, "Selection of cluster prototypes from data by a genetic algorithm," in *Proc. 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Alemanha*, 1997, pp. 1683–1688.

[9] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 8, pp. 841–847, 1991.

[10] E. E. Korkmaz, J. Du, R. Alhajj, and K. Barker, "Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering," *Intelligent Data Analysis*, vol. 10, no. 2, pp. 163–182, 2006.

[11] D. Dutta, P. Dutta, and J. Sil, "Simultaneous feature selection and clustering with mixed features by multi objective genetic algorithm," *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 1, pp. 41–54, 2014.

[12] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 991–1005, 2009.

[13] K. S. N. Ripon, C.-H. Tsang, S. Kwong, and M.-K. Ip, "Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 1200–1203.

[14] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary Computation*, 2014.

[15] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. P. L. F. De Carvalho, "A survey of evolutionary algorithms for clustering," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 2, pp. 133–155, 2009.

[16] S. Bickel and T. Scheffer, "Multi-view clustering." in *ICDM*, vol. 4, 2004, pp. 19–26.

[17] A. Wahid, X. Gao, and P. Andreae, "Exploiting user queries for search result clustering," in *Web Information Systems Engineering–WISE 2013*. Springer, 2013, pp. 111–120.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[19] F. Gullo, C. Domeniconi, and A. Tagarelli, "Projective clustering ensembles," *Data Mining and Knowledge Discovery*, vol. 26, no. 3, pp. 452–511, 2013.

[20] N. Nguyen and R. Caruana, "Consensus clusterings," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 607–612.

[21] D. Crabtree, X. Gao, and P. Andreae, "Improving web clustering by cluster selection," in *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. IEEE, 2005, pp. 172–178.

[22] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[23] N. Iam-on and S. Garrett, "Linkclue: A matlab package for link-based cluster ensembles," *Journal of Statistical Software*, vol. 36, no. i09, 2010.

[24] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, pp. 80–83, 1945.