applied sciences

MDPI

Article

An Improved Evolutionary Multi-Objective Clustering Algorithm Based on Autoencoder

Mingxin Qiu [1], Yingyao Zhang [1,*], Shuai Lei [1] and Miaosong Gu [2]

[1] School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China; 2233091@tongji.edu.cn (M.Q.); 2331810@tongji.edu.cn (S.L.)
[2] Economic Research Institute of State Grid Zhejiang Electric Power Company, Hangzhou 102209, China; gumiaosong@ncepu.edu.cn
* Correspondence: zhangyingyao@tongji.edu.cn

Abstract: Evolutionary multi-objective clustering (EMOC) algorithms have gained popularity recently, as they can obtain a set of clustering solutions in a single run by optimizing multiple objectives. Particularly, in one type of EMOC algorithm, the number of clusters k is taken as one of the multiple objectives to obtain a set of clustering solutions with different k. However, the numbers of clusters k and other objectives are not always in conflict, so it is impossible to obtain the clustering solutions with all different k in a single run. Therefore, evolutionary multi-objective k-clustering (EMO-KC) has recently been proposed to ensure this conflict. However, EMO-KC could not obtain good clustering accuracy on high-dimensional datasets. Moreover, EMO-KC's validity is not ensured as one of its objectives (SSD_{exp}, which is transformed from the sum of squared distances (SSD)) could not be effectively optimized and it could not avoid invalid solutions in its initialization. In this paper, an improved evolutionary multi-objective clustering algorithm based on autoencoder (AE-IEMOKC) is proposed to improve the accuracy and ensure the validity of EMO-KC. The proposed AE-IEMOKC is established by combining an autoencoder with an improved version of EMO-KC (IEMO-KC) for better accuracy, where IEMO-KC is improved based on EMO-KC by proposing a scaling factor to help effectively optimize the objective of SSD_{exp} and introducing a valid initialization to avoid the invalid solutions. Experimental results on several datasets demonstrate the accuracy and validity of AE-IEMOKC. The results of this paper may provide some useful information for other EMOC algorithms to improve accuracy and convergence.

Keywords: multi-objective clustering; autoencoder; deep learning; high-dimensional datasets

Citation: Qiu, M.; Zhang, Y.; Lei, S.; Gu, M. An Improved Evolutionary Multi-Objective Clustering Algorithm Based on Autoencoder. *Appl. Sci.* **2024**, *14*, 2454. https://doi.org/10.3390/app14062454

Academic Editor: Chilukuri K. Mohan

Received: 12 February 2024
Revised: 4 March 2024
Accepted: 7 March 2024
Published: 14 March 2024

1. Introduction

Clustering is one of the most important tasks in data mining and machine learning, which is commonly used in pattern analysis, customer segmentation, image segmentation, and other fields [1]. The purpose of clustering is to divide a dataset into different clusters, to better understand the characteristics of the dataset, discover hidden rules and relationships among data points within each cluster, and carry out subsequent analysis and decisions. For traditional clustering algorithms, the number of clusters k needs to be determined in advance, which would have a significant impact on the final clustering performance [1–3]. However, it would be difficult to select an appropriate k without prior knowledge of the dataset. A common approach is to select the optimal k based on the clustering results through an enumeration method. This approach is simple to implement but requires multiple runs, which is obviously deficient especially when the size of the dataset or the range of k is large.

For this issue, evolutionary multi-objective clustering (EMOC) algorithms have gained popularity, as they can obtain a set of clustering solutions in a single run by optimizing multiple objectives [2–6]. Particularly, for one type of EMOC algorithm, the number of

clusters k is taken as one of the multiple objectives to obtain a set of clustering solutions with different k. However, the numbers of clusters k and other objectives are not always in conflict, so clustering solutions with all different k cannot be obtained in a single run [6]. In this case, evolutionary multi-objective k-clustering (EMO-KC) has been proposed recently, which has an effective bi-objective model to ensure this conflict [7]. In this model, the number of clusters k and SSD_{exp} (see Equation (5)), which is transformed from the sum of squared distances (SSD), are taken as two objectives to ensure the conflict. The advantages of EMO-KC have been demonstrated in CCDG-K [8].

However, there are still several limitations of EMO-KC in its accuracy and validity. EMO-KC usually has a large number of decision variables, which increases with the dimensionality of the datasets, resulting in a large search space [9]. It is difficult for EMO-KC to converge to the global optimal solutions in such a large search space. As a result, the clustering accuracy of EMO-KC on high-dimensional datasets is limited. Furthermore, one of the two objectives of EMO-KC, SSD_{exp}, could not be effectively optimized. If SSD is large, the first term of SSD_{exp} (see Equation (5)), $1 - \exp^{-1 \cdot SSD}$, would be approximately equal to 1, and the second term of SSD_{exp}, $-k$, would dominate SSD_{exp}. Thus, different clustering solutions with the same k in the population will obtain almost the same SSD_{exp}, making it difficult to optimize the objective of SSD_{exp}. As a result, EMO-KC's validity is limited. In addition, the treatment of invalid solutions is not considered in the initialization process of EMO-KC. Points in the search space are randomly selected as the cluster centroids and encoded into the chromosomes representing the clustering solutions in EMO-KC. Thus, there may be some invalid clusters without any data points in the clustering solution for a certain k, making the solution unrealistic as its number of valid clusters is less than k. As a result, it is not ensured to obtain the clustering solutions with all different k, which also limits the validity of EMO-KC.

Previous studies [8–13] have focused on the above issues of EMO-KC. A reduced-length chromosome encoding method was used to reduce the number of decision variables in [9], which was also commonly used in earlier studies to improve the clustering accuracy, especially on high-dimensional datasets [10–12]. The number of features representing the dimensionality of the datasets was taken as an optimizing objective in [13] to reduce the number of decision variables. However, the dimension reduction of the input datasets has rarely been considered. For this issue, autoencoder as a data dimension reduction method based on deep learning has gained popularity in clustering and helps to obtain good clustering accuracy especially on high-dimensional datasets [14–20]. It can maintain the nonlinear feature of the datasets while reducing the dimensionality. Song et al. [14] used the autoencoder to reduce the dimensionality of the datasets by mapping the datasets to the low-dimensional embedding layer of the autoencoder as the feature representation. Then, the feature representation was clustered by k-means, which could significantly improve the accuracy of clustering. The following studies have focused on the expressions of the autoencoder's loss functions and the clustering algorithms used [15–20]. Thus, EMO-KC as a clustering algorithm is expected to obtain better clustering accuracy when combined with the autoencoder. However, the combination of EMO-KC and the autoencoder has not been proposed in related studies. Furthermore, Zhu et al. [9] analyzed the invalidity of EMO-KC, i.e., the objective of SSD_{exp} could not be effectively optimized. However, no measures were taken to address this issue in their study and other relevant studies. In addition, to reduce the influence of the invalid clusters, a constrained decomposition based on grids (CDG) was introduced into CCDG-K [8] to divide the clustering task into multiple subtasks, each of which focused on optimizing the single objective of SSD_{exp}. This ensured that clustering solutions could be obtained for all different k. However, the treatment of the invalid solutions was still not considered in CCDG-K. Actually, an effective approach to avoid the invalid solutions is to select data points as the cluster centroids and encode them into the chromosome during the initialization process (called valid initialization for short) as in GKA [21] and MOKGA [22]. However, valid initialization has not been considered in EMO-KC yet.

In this paper, an improved evolutionary multi-objective clustering algorithm based on autoencoder (AE-IEMOKC) is proposed to improve the accuracy and ensure the validity of EMO-KC. The proposed AE-IEMOKC is established by combining an autoencoder with an improved version of EMO-KC (IEMO-KC) for better accuracy, where IEMO-KC is improved based on EMO-KC by proposing a scaling factor to help effectively optimize the objective of SSD_{exp} and introducing valid initialization to avoid the invalid solutions. The accuracy and validity of AE-IEMOKC are demonstrated on several datasets. The results of this paper may provide some useful information for other EMOC algorithms to improve accuracy and convergence.

2. Proposed Algorithm

Figure 1 shows the architecture of the proposed AE-IEMOKC, which is established by combining an autoencoder with IEMO-KC. First, the original dataset X is mapped to the low-dimensional embedding layer as the feature representation H of X by the encoder of the autoencoder. Then, H is transformed as the reconstructed data X' by the decoder of the autoencoder. This process is repeated iteratively to minimize the loss (see Equation (4)). Then, the final H obtained from the embedding layer is divided into different clusters by IEMO-KC. The clusters are constantly adjusted during the continuous iterations of IEMO-KC to minimize the two objectives f_1 and f_2 (see Equation (5)) for better clustering solutions. A set of optimal non-dominated clustering solutions with different k (Pareto front) can be obtained after the number of generations (*gen*) reaches the maximum (*maxgen*). The following subsections provide a detailed introduction to the autoencoder and IEMO-KC.

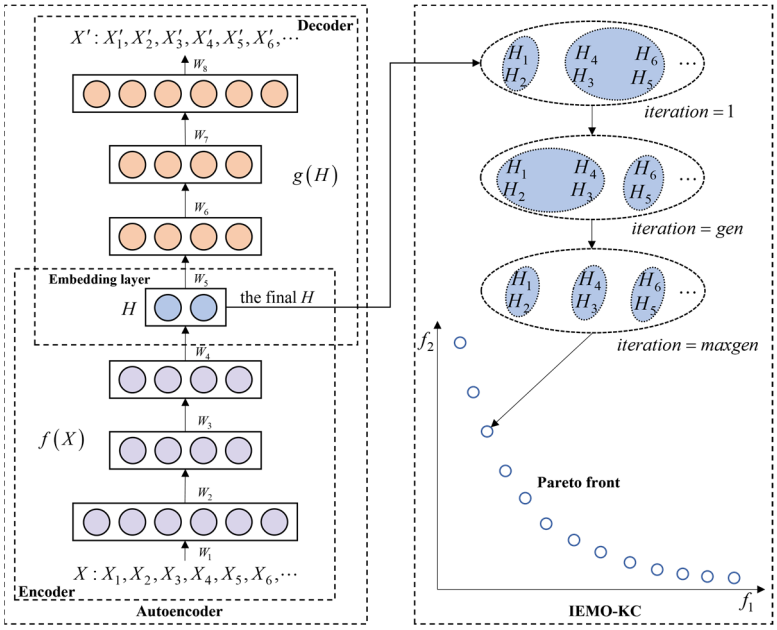


Figure 1. Architecture of the proposed AE-IEMOKC.

2.1. Autoencoder (AE)

Autoencoder is the part of AE-IEMOKC responsible for dimension reduction of the input datasets. It is a neural network based on deep learning and consists of an encoder and a decoder. The eight-layer neural network shown in Figure 1 is taken as an example. The encoder focuses on mapping the original dataset X to the low-dimensional embedding layer as the feature representation H of X through three hidden layer and one linear layer sequentially, which can be defined as a transformation:

$$H = f(X) = W_4^T \phi\left(W_3^T \phi\left(W_2^T \phi\left(W_1^T X\right)\right)\right) \quad (1)$$

where $\phi(\cdot)$ is a ReLU activation function [23] and $\phi(\cdot) = \max(0, X)$; W_1, W_2, W_3, and W_4 are the weights of the encoder network. For simplicity, the bias term b_i for each layer in the formulation is dropped. The decoder focuses on transforming H to the reconstructed data X' through three hidden layers and one linear layer sequentially, which can also be defined as a transformation:

$$X' = g(H) = W_8^T \phi\left(W_7^T \phi\left(W_6^T \phi\left(W_5^T H\right)\right)\right) \tag{2}$$

where W_5, W_6, W_7, and W_8 are the wights of the decoder network. The autoencoder is able to learn the nonlinear feature of X by minimizing the reconstruction loss $L_{\text{rec}}(X)$:

$$L_{\text{rec}}(X) = \frac{1}{m}\sum_{i=1}^{m}\left\|X_i - X'_i\right\|^2 \tag{3}$$

where m is the amount of the original dataset X. However, minimizing the reconstruction loss $L_{\text{rec}}(X)$ contributes little to clustering [14]. Thus, a clustering loss $L_{\text{cl}}(H)$ is considered together with the reconstruction loss $L_{\text{rec}}(X)$, and the whole loss $L(X, H)$ is defined as follows:

$$\begin{gathered} L(X,H) = L_{\text{rec}}(X) + \lambda L_{\text{cl}}(H) \\ \text{where } L_{\text{cl}}(H) = \left(\frac{10}{m \cdot d^*}\sum_{r=1}^{k^*}\sum_{H_i \in C_r}\|H_i - m_r\|^2\right)^2 \\ m_r = \left(m_r^1, m_r^2, \ldots, m_r^{d^*}\right) \end{gathered} \tag{4}$$

where $\lambda \geq 0$ is a parameter to balance the reconstruction loss and the clustering loss, k* is the actual number of clusters of X, d* is the dimensionality of the embedding layer, $m_r = \left(m_r^1, m_r^2, \ldots, m_r^{d^*}\right)$ denotes the rth cluster centroid of H, and C_r denotes the collection of H in the rth cluster. By minimizing the loss $L(X, H)$, the autoencoder is able to obtain the low-dimensional final H, which maintains the nonlinear feature of X and is suitable for clustering.

2.2. IEMO-KC

IEMO-KC is the other part of AE-IEMOKC responsible for clustering, which divides the final H into different clusters. This division has multiple schemes, representing multiple clustering solutions. In IEMO-KC, a bi-objective model is used to evaluate these solutions, and an optimizer is used to select the solutions with better evaluation results to optimize the two objectives in the bi-objective model. This subsection introduces the bi-objective model and the optimizer, as well as the chromosome encoding method used to represent the solutions.

2.2.1. Bi-Objective Model

The bi-objective model of EMO-KC can be represented as follows [7]:

$$\begin{gathered} \text{Min } F(H) = \left\{f_1(H) = \text{SSD}_{\text{exp}}, f_2(H) = k\right\} \\ \text{where SSD}_{\text{exp}} = (1 - \exp^{-1 \cdot \text{SSD}}) - k \\ \text{SSD} = \sum_{r=1}^{k}\sum_{H_i \in C_r}\|H_i - m_r\|^2 \\ m_r = \left(m_r^1, m_r^2, \ldots, m_r^{d^*}\right) \end{gathered} \tag{5}$$

where SSD_{exp} and the number of clusters k are taken as two objectives to minimize, and SSD_{exp} is transformed from SSD, which is the squared sum of the distance from the data point to its cluster centroid. However, if the SSD in SSD_{exp} is large, the $-k$ in SSD_{exp} will dominate the SSD_{exp} and different solutions with the same k will obtain almost the same evaluation results, making it difficult to distinguish better solutions to optimize the

objective of SSD_{exp}. Thus, in this paper, a scaling factor is proposed to scale the SSD to an appropriate range so that the bi-objective model used in IEMO-KC can be represented as follows:

$$\begin{gathered} \text{Min } F(H) = \{f_1(H) = \text{SSD}_{\text{exp}}, f_2(H) = k\} \\ \text{where SSD}_{\text{exp}} = (1 - \exp^{-\alpha \cdot \text{SSD}}) - k \\ \text{SSD} = \sum_{r=1}^{k} \sum_{H_i \in C_r} \|H_i - m_r\|^2 \\ m_r = \left(m_r^1, m_r^2, \ldots, m_r^{d^*}\right) \end{gathered} \tag{6}$$

where α is the scaling factor that varies for different datasets.

2.2.2. Optimizer

NSGA-II [24] is employed as the optimizer due to its simpleness. It is slightly adjusted in IEMO-KC, and its pseudo-code is shown in Algorithm 1.

Algorithm 1: NSGA-II for IEMO-KC.

Input: Maximum generation *maxgen*, population size N, a range of k
Output: A set of optimal ono-dominated solutions with different k, Pareto front
1: Initialize a set of N random parent solutions, *PS*
2: Assign to each solution with a random different k
3: While $gen \leq maxgen$
4: Generate N offspring solutions *OS* by crossover and mutation operators
5: Combine *PS* and *OS* together to form *jointS*
6: Evaluate *jointS* by the fast non-dominated sorting approach and the crowding distance [24]
7: Select the best N solutions from *jointS* to form the new parent *PS*
8: $gen \leftarrow gen + 1$
9: End while
10: Select a set of optimal non-dominated solutions with different k from *PS* to form the Pareto front

The algorithm generates a set of N random initialized parent solutions (*PS*) and assigns each solution with a random different k before the iterations. During each iteration, the same number of offspring solutions (*OS*) is generated from *PS* through simulated binary crossover (SBX) and polynomial mutation (PM) [24]. Specifically, two offspring solutions are generated from two randomly selected parent solutions by using the SBX operator, with appropriate values of the probability of applying recombination (p_c) and the magnitude of the expected variation from the parent values (η_c). Note that the k values of the two offspring solutions are inherited from the two parent solutions. Subsequently, a new solution generated by using the SBX operator is further mutated by using the PM operator, with appropriate values of the probability of applying mutation (p_m) and a mutation distribution parameter (η_m). Note that the k of this solution remains unchanged in the mutation process. Then, *PS* and *OS* are combined as *jointS*, which is then evaluated by the fast non-dominated sorting approach and the crowding distance. The best N solutions from *jointS* are then selected to form the new *PS*. After the iterations, a set of optimal non-dominated solutions with different k can be selected from *PS* to form the Pareto front.

2.2.3. Chromosome Encoding Method

The centroid-based chromosome encoding method is used, where the chromosome is composed of the cluster centroids. To avoid invalid solutions, it is based on valid initialization in this paper. Specifically, different k_{max} data points in the final H are randomly selected as the cluster centroids and encoded into the chromosome, where k_{max} is the maximum k, and the default range of k is [2, k_{max}]. The length of each chromosome is unified as $n = d^* \cdot k_{max}$. Figure 2 shows the centroid-based chromosome encoding method based on the valid initialization using 10 two-dimensional data points as an example. When k_{max} is set to 4, four data points are selected as the cluster centroids $c = \{c_1, c_2, c_3, c_4\}$ and

encoded into the chromosome. After the initialization, each chromosome is assigned with a random k. If the random k of a chromosome is 2, only (0.1, 0.7, 0.3, 0.3) will be taken as the decision variables of this chromosome.

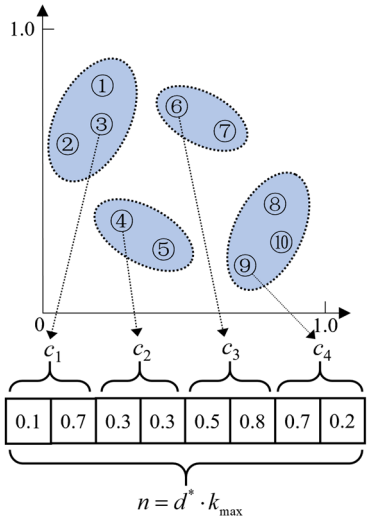


Figure 2. Schematic diagram of the centroid-based chromosome encoding method based on valid initialization.

3. Experimental Settings

3.1. Datasets and Evaluation Metrics

Five real datasets from UCI at https://archive.ics.uci.edu/ (accessed on 11 February 2024) are used in the experiments, as shown in Table 1. Each dataset has a high dimensionality, except for the Iris dataset. Normalization of the original data or the final H is used before input to the autoencoder or IEMO-KC. Two standard unsupervised evaluation metrics are used to evaluate the clustering accuracy, Adjusted Rand Index (ARI) [25] and Clustering Accuracy (ACC) [26]. ARI would range from -1 to 1, while ACC would range from 0 to 1. Higher ARI and ACC indicate better accuracy. The two metrics have their own advantages and disadvantages, but analysis based on their combination is effective [17]. Furthermore, if the objective of f_1 (see Equation (6)) could be effectively optimized, smaller f_1 and SSD could be obtained simultaneously. Thus, SSD and f_1 are used to evaluate validity. Smaller SSD and f_1 indicate better validity.

Table 1. Summary of experimental datasets.

Dataset	Amount	Dimensionality	Actual Number of Clusters
Iris	150	4	3
Wine	214	9	3
Seeds	210	7	3
Breast Cancer Wisconsin (BCW)	569	30	2
Optdigits	1797	64	10

3.2. Parameter Settings

The structural settings of the autoencoder used in the experiments are consistent across all the datasets. The dimension of the encoder network is set to d-500-500-2000-d*, where d is the dimensionality of the input dataset and d* is the dimensionality of the embedding layer. d* is set to 3 in this paper. This means that the input dataset will be transformed to the 3-dimensional final H. The decoder is a mirrored version of the encoder. All layers of the network are fully connected. Except for the layer before the embedding layer and the layer before the output layer, each layer applies a ReLU activation function before being fed

to the next layer. The autoencoder is trained for each dataset using the Adam optimizer [27] with different learning rates. The parameter λ also varies for different datasets. The settings of the learning rate and the parameter λ are shown in Table 2. The autoencoder is pre-trained for 100 iterations to minimize the reconstruction loss (see Equation (3)), and further fine-tuned for 200 iterations to minimize the whole loss (see Equation (4)). In this paper, the clustering loss in Equation (4) is obtained by k-means [28].

Table 2. Settings of the learning rate and the parameter λ for different datasets.

Dataset	Learning Rate	λ
Iris	1×10^{-3}	1×10^{-3}
Wine	1×10^{-3}	5×10^{-2}
Seeds	5×10^{-5}	1×10^{1}
BCW	7×10^{-4}	5×10^{-2}
Optdigits	1×10^{-3}	1×10^{-2}

For IEMO-KC, the *maxgen* is set to 500 for all the datasets. The range of k is set to [2,15] in this paper, although the $k_{\max}$ can be set to a larger value. The population size N is set to 100. p_c and η_c of SBX are set to 1 and 15, respectively. p_m and η_m are set to $1/(d^* \cdot k_{\max})$ and 20, respectively. The scaling factor α is set to $10/(m \cdot d^*)$, where m is the amount of each dataset.

4. Results and Discussions

4.1. The Accuracy of AE-IEMOKC

The accuracy of AE-IEMOKC is demonstrated by comparing with EMO-KC [7], GKA [21], and MOKGA [22]. Furthermore, the population size N, and the *maxgen*, crossover, and mutation operators for EMO-KC, GKA, and MOKGA are kept consistent with those in Section 3.2. Values of p_m for EMO-KC, GKA, and MOKGA are set to $1/(d \cdot k_{\max})$ as the input datasets for them are the original datasets instead of the final H.

To briefly demonstrate the clustering accuracy of AE-IEMOKC, a solution corresponding to the actual number of clusters of each dataset is selected from the set of clustering solutions obtained as an example for explanation. Table 3 shows the obtained ARI and ACC. It can be clearly observed that EMO-KC has the smallest ARI and ACC on all the datasets due to its invalidity. It is evident that AE-IEMOKC has the highest ARI and ACC on all the datasets, which means that the best accuracy solutions can be obtained by our proposed algorithm. Specifically, the largest improvement in AE-IEMOKC's clustering accuracy over GKA and MOKGA is on the Optdigits dataset. It is difficult for both GKA and MOKGA to converge on the dataset Optdigits due to the high dimensionality, which results in numerous decision variables and a large search space. However, AE-IEMOKC converges easily due to the autoencoder's ability to reduce the dimensionality.

Table 3. The ARI and ACC of the clustering results obtained by EMO-KC, GKA, MOKAG, and AE-IEMOKC on the five datasets under their actual number of clusters.

Metric	Algorithm	Iris	Wine	Seeds	BCW	Optdigits
ARI	EMO-KC	0.51	0.29	0.47	0.04	0.20
	GKA	0.72	0.85	0.70	0.73	0.37
	MOKGA	0.72	0.85	0.70	0.73	0.37
	AE-IEMOKC	0.87	0.90	0.77	0.79	0.67
ACC	EMO-KC	0.67	0.64	0.74	0.66	0.38
	GKA	0.89	0.95	0.89	0.93	0.57
	MOKGA	0.89	0.95	0.89	0.93	0.57
	AE-IEMOKC	0.95	0.97	0.92	0.95	0.79

Figure 3 shows the clustering results obtained by GKA, MOKGA, and AE-IEMOKC on the Iris, Wine, and Seeds datasets when $k = 3$, where the clustering results obtained by EMO-KC are not shown due to their invalidity. The mark (+) denotes the cluster centroid. It can be observed that the distribution of the data points in AE-IEMOKC is significantly different from that in GKA and MOKGA, as the data points in AE-IEMOKC are actually the final H transformed from the original datasets by the autoencoder. Since the final H is more suitable for clustering, AE-IEMOKC is able to achieve better clustering results, as it can result in tighter data points within the same cluster and clearer distinctions between data points within different clusters. Similar results are also observed for the BCW and Optdigits datasets.

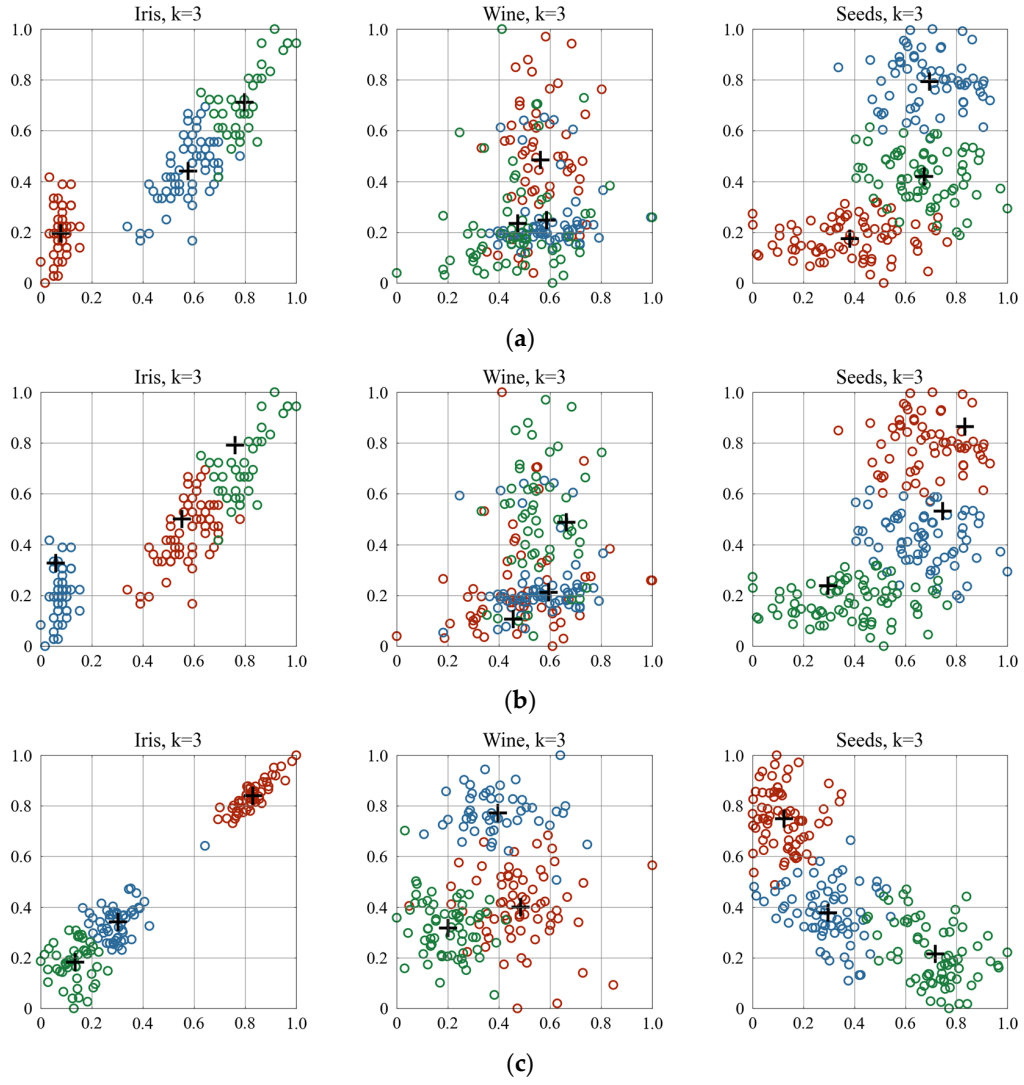


Figure 3. Clustering results obtained by GKA, MOKGA, and AE-IEMOKC on the Iris, Wine, and Seeds datasets when $k = 3$: (**a**) The clustering results obtained by GKA (**b**) The clustering results obtained by MOKGA. (**c**) The clustering results obtained by AE-IEMOKC.

4.2. The Validity of AE-IEMOKC

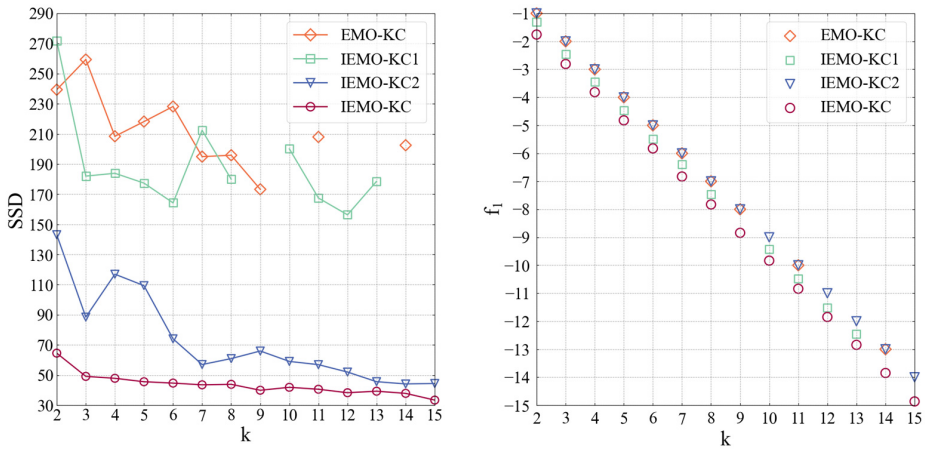
The validity of AE-IEMOKC is ensured by its IEMO-KC part. To demonstrate the validity of AE-IEMOKC, this subsection makes a comparison among EMO-KC, IEMO-KC1 (EMO-KC based on the scaling factor), IEMO-KC2 (EMO-KC based on the valid initialization), and IEMO-KC, without considering the autoencoder. Note that the values of p_{m} for the four algorithms are set to $1/(d \cdot k_{\max})$, and α for both IEMO-KC1 and IEMO-KC is set to $10/(m \cdot d)$, since the input datasets for them are the original datasets.

Table 4 shows the SSD and f_1 obtained by the four algorithms on the five datasets under their actual number of clusters. It shows that the SSD of EMO-KC is the largest, and the f_1 of EMO-KC is approximately equal to -2.00, which shows the poor validity of EMO-KC. However, the SSD and f_1 of IEMO-KC1 are smaller than those of EMO-KC, which suggests that the scaling factor can help effectively optimize the objective of SSD_{exp}. It can also be observed that the SSD of IEMO-KC2 is smaller than those of EMO-KC and IEMO-KC1. This is because in the valid initialization of IEMO-KC2, data points rather than points in the search space are selected as the cluster centroids, which allows the cluster centroids to be closer to the other data points, resulting in a smaller SSD. However, the f_1 of IEMO-KC2 is approximately equal to -2.00 due to the lack of the scaling factor, indicating that the objective of SSD_{exp} has not been effectively optimized. However, it is evident that IEMO-KC is able to obtain the smallest SSD and f_1 simultaneously, which shows that the combination of the scaling factor and valid initialization in IEMO-KC contributes to the validity.

Table 4. The SSD and f_1 of the clustering results obtained by EMO-KC, IEMO-KC1, IEMO-KC2, and IEMO-KC on the five datasets under their actual number of clusters.

Metric	Algorithm	Iris	Wine	Seeds	BCW	Optdigits
SSD	EMO-KC	34.89	259.38	145.79	2507.83	23,678.61
	IEMO-KC1	31.10	182.02	128.60	1902.68	23,646.44
	IEMO-KC2	10.09	88.47	117.32	423.08	9137.39
	IEMO-KC	6.98	49.00	22.03	216.25	7488.99
f_1	EMO-KC	-2.00	-2.00	-2.00	-2.00	-2.00
	IEMO-KC1	-2.60	-2.46	-2.42	-2.33	-2.13
	IEMO-KC2	-2.00	-2.00	-2.00	-2.00	-2.00
	IEMO-KC	-2.89	-2.81	-2.86	-2.88	-2.52

The validity is further demonstrated using the Wine dataset as an example. Figure 4 shows the Pareto fronts of SSD and f_1 obtained by the four algorithms. It can be observed that IEMO-KC2 and IEMO-KC are able to obtain clustering solutions with all different k. This indicates that valid initialization is able to avoid invalid solutions. It can also be clearly observed that all the solutions obtained by EMO-KC, IEMO-KC1, and IEMO-KC2 are Pareto-dominated by those obtained by IEMO-KC, which further shows that the validity is ensured by the combination of the scaling factor and valid initialization. Similar results are observed for other datasets.



(**a**) (**b**)

Figure 4. Pareto fronts obtained by the four algorithms on the Wine dataset, where solutions containing the invalid clusters are removed due to their invalidity: (**a**) Comparison of SSD. (**b**) Comparison of f_1.

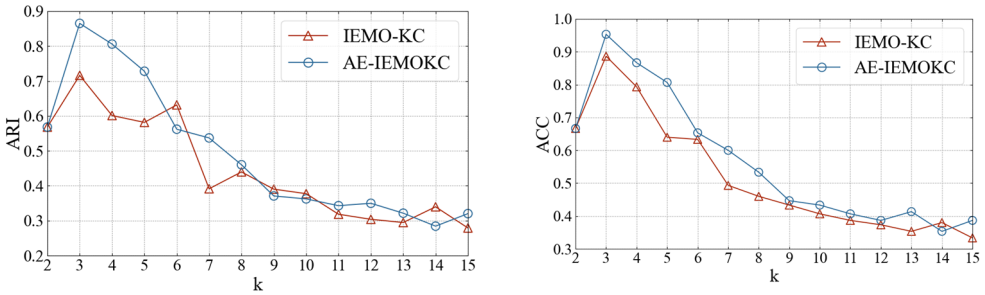
4.3. The Influence of the Autoencoder

The influence of the autoencoder is further discussed by comparing AE-IEMOKC with IEMO-KC. Note that p_m and α for IEMO-KC are set to $1/(d \cdot k_{max})$ and $10/(m \cdot d)$, respectively, since the input datasets for IEMO-KC are the original datasets.

Table 5 shows the ARI and ACC of IEMO-KC and AE-IEMOKC on the five datasets under their actual number of clusters. It can be observed that AE-IEMOKC is able to obtain higher ARI and ACC, especially on the Optdigits dataset. This directly demonstrates that the autoencoder can improve the clustering accuracy due to its ability to obtain the feature representation of the dataset suitable for clustering and its ability to reduce the dimensionality of the dataset. In fact, this improvement is not limited to the solution with the actual number of clusters. Taking the Iris dataset as an example, Figure 5 shows the ARI and ACC of the Pareto fronts obtained by IEMO-KC and AE-IEMOKC. It can be observed that the improvement in the accuracy of the autoencoder also works for some other solutions. However, it does not work for all the solutions, as the clustering loss in Equation (4) is obtained under the actual number of clusters. Overall, the autoencoder has a significant positive effect on the solutions, which have numbers of clusters close to the actual number of clusters.

Table 5. The ARI and ACC of the clustering results obtained by IEMO-KC and AE-IEMOKC on the five datasets under their actual number of clusters.

Metric	Algorithm	Iris	Wine	Seeds	BCW	Optdigits
ARI	IEMO-KC	0.72	0.87	0.70	0.72	0.33
	AE-IEMOKC	0.87	0.90	0.77	0.79	0.67
ACC	IEMO-KC	0.89	0.96	0.89	0.92	0.48
	AE-IEMOKC	0.95	0.97	0.92	0.95	0.79



(**a**) (**b**)

Figure 5. The ARI and ACC of the Pareto fronts obtained by IEMO-KC and AE-IEMOKC on the Iris dataset: (**a**) Comparison of ARI. (**b**) Comparison of ACC.

Figure 6 shows the clustering results obtained by IEMO-KC and AE-IEMOKC on the Iris dataset when k = 2, 3, 4. It can be clearly observed that AE-IEMOKC is able to achieve better clustering results under the actual number of clusters and its neighboring number of clusters, as the final H transformed from the original dataset by the autoencoder is more suitable for clustering. Similar results are observed for the other datasets.

Figure 7 shows the running time averaged over 10 runs of EMO-KC, IEMO-KC1, IEMO-KC, and AE-IEMOKC on the five datasets. Each algorithm is implemented in Python 3.9 with a computer configuration of AMD R7-5800H CPU, 16 GB RAM, and RTX3050 4 GB GPU. It is observed that IEMO-KC1 consumes slightly more time than EMO-KC due to the additional computation of the scaling factor. However, the valid initialization has a greater influence than the scaling factor as IEMO-KC consumes significantly more time than IEMO-KC1. In fact, more time is not mainly consumed in the initialization process, but in the optimization, as the ensured validity of IEMO-KC makes the optimization more effective and thus more complex. It is observed that AE-IEMOKC consumes more time

than IEMO-KC on the Iris, Wine, Seeds, and BCW datasets as the autoencoder part of AE-IEMOKC takes a lot of time to be pre-trained and fine-tuned, which greatly reduces the efficiency of AE-IEMOKC. However, the IEMO-KC part of AE-IEMOKC converges faster than IEMO-KC on the five datasets due to the autoencoder's ability to reduce the dimensionality, which is obvious on the BCW and Optdigits datasets. This suggests the autoencoder part of AE-IEMOKC is able to accelerate the convergence.

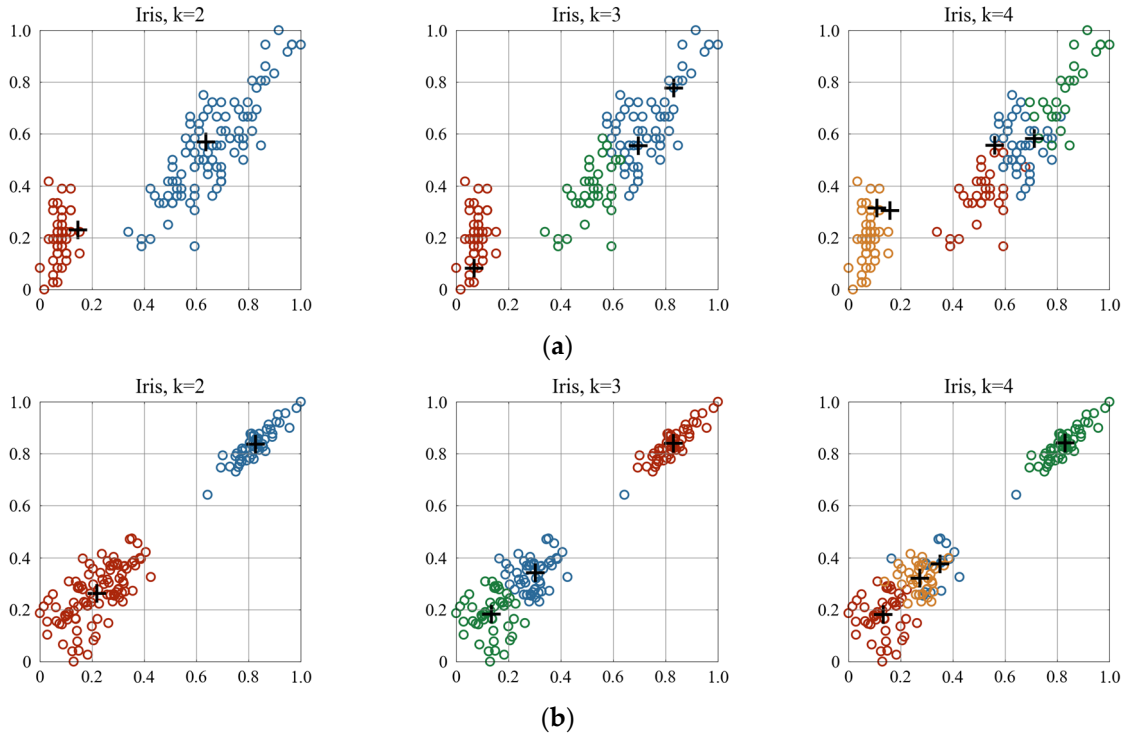


Figure 6. The clustering results obtained by IEMO-KC and AE-IEMOKC on the Iris dataset: (**a**) The clustering results obtained by IEMO-KC. (**b**) The clustering results obtained by AE-IEMOKC.

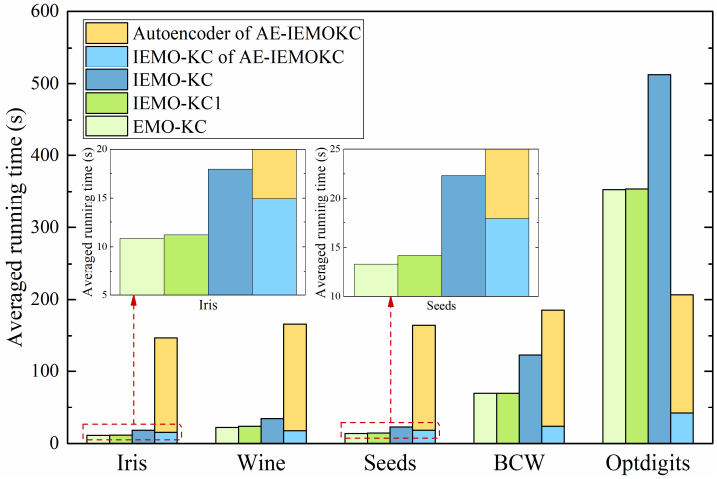


Figure 7. Averaged running time of EMO-KC, IEMO-KC1, IEMO-KC, and AE-IEMOKC on the five datasets.

5. Conclusions

In this paper, an improved evolutionary multi-objective clustering algorithm based on autoencoder (AE-IEMOKC) was proposed to improve the accuracy and ensure the validity of evolutionary multi-objective optimization k-clustering (EMO-KC). The proposed

AE-IEMOKC was established by combining an autoencoder with an improved version of EMO-KC (IEMO-KC) for better accuracy, where IEMO-KC was improved based on EMO-KC by proposing a scaling factor to help effectively optimize the objective of SSD_{exp} and introducing valid initialization to avoid invalid solutions. The accuracy and validity of AE-IEMOKC were demonstrated on several datasets. The results showed that the proposed AE-IEMOKC could obtain good accuracy on high-dimensional datasets. It was also shown that the scaling factor could help effectively optimize the objective of SSD_{exp}, the valid initialization could avoid the invalid solutions, and the combination of them could ensure the validity. Furthermore, the autoencoder part of AE-IEMOKC was shown to improve the accuracy and accelerate the convergence due to its ability to obtain the feature representation of the dataset suitable for clustering and its ability to reduce the dimensionality of the dataset, which may provide some useful information for other EMOC algorithms to improve accuracy and convergence. Future research includes improving the accuracy of the solutions with all different k obtained by AE-IEMOKC and improving the efficiency of AE-IEMOKC.

Author Contributions: Conceptualization, M.Q. and Y.Z.; methodology, M.Q., Y.Z. and S.L.; software, M.Q.; validation, M.Q., Y.Z. and S.L.; formal analysis, M.Q.; investigation, M.Q.; resources, M.Q.; data curation, M.Q.; writing—original draft preparation, M.Q. and Y.Z.; writing—review and editing, M.Q., Y.Z., S.L. and M.G.; visualization, M.Q.; supervision, Y.Z.; project administration, M.G.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, grant number 2022YFB3305802.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://archive.ics.uci.edu/ (accessed on 11 February 2024).

Acknowledgments: We are grateful to the anonymous reviewers for their comments on this manuscript.

Conflicts of Interest: Author Miaosong Gu was employed by the Economic Research Institute of State Grid Zhejiang Electric Power Company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Liu, C.; Liu, J.; Peng, D.; Wu, C. A general multiobjective clustering approach based on multiple distance measures. *IEEE Access* **2018**, *6*, 41706–41719. [CrossRef]
2. Mukhopadhyay, A.; Maulik, U.; Bandyopadhyay, S. A survey of multiobjective evolutionary clustering. *ACM Comput. Surv.* **2015**, *47*, 1–46. [CrossRef]
3. Abu Khurma, R.; Aljarah, I. A review of multiobjective evolutionary algorithms for data clustering problems. In *Evolutionary Data Clustering: Algorithms and Applications*; Springer: Singapore, 2021; pp. 177–199.
4. Hruschka, E.R.; Campello, R.J.; Freitas, A.A. A survey of evolutionary algorithms for clustering. *IEEE Trans. Syst. Man Cybern. Part C* **2009**, *39*, 133–155. [CrossRef]
5. Bong, C.W.; Rajeswari, M. Multiobjective clustering with metaheuristic: Current trends and methods in image segmentation. *IET Image Process.* **2012**, *6*, 1–10. [CrossRef]
6. Morimoto, C.Y.; Pozo, A.; de Souto, M.C. A Review of Evolutionary Multi-objective Clustering Approaches. *arXiv* **2021**, arXiv:2110.08100.
7. Wang, R.; Lai, S.; Wu, G.; Xing, L.; Wang, L.; Ishibuchi, H. Multi-clustering via evolutionary multi-objective optimization. *Inf. Sci.* **2018**, *450*, 128–140. [CrossRef]
8. Wang, L.; Cui, G.; Zhou, Q.; Li, K. A multi-clustering method based on evolutionary multiobjective optimization with grid decomposition. *Swarm Evol. Comput.* **2020**, *55*, 100691. [CrossRef]
9. Zhu, S.; Xu, L.; Goodman, E.D. Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. *Knowl. Based Syst.* **2020**, *188*, 105018. [CrossRef]
10. Garza-Fabre, M.; Handl, J.; Knowles, J. An improved and more scalable evolutionary approach to multiobjective clustering. *IEEE Trans. Evol. Comput.* **2017**, *22*, 515–535. [CrossRef]

11. Zhu, S.; Xu, L.; Cao, L. A study of automatic clustering based on evolutionary many-objective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Kyoto, Japan, 15–19 July 2018.
12. Bechikh, S.; Elarbi, M.; Hung, C.C.; Hamdi, S.; Said, L.B. A Hybrid Evolutionary Algorithm with Heuristic Mutation for Multi-objective Bi-clustering. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019.
13. Di Nuovo, A.G.; Palesi, M.; Catania, V. Multi-objective evolutionary fuzzy clustering for high-dimensional problems. In Proceedings of the 2007 IEEE International Fuzzy Systems Conference, London, UK, 23–26 July 2007.
14. Song, C.; Liu, F.; Huang, Y.; Wang, L.; Tan, T. Auto-encoder based data clustering. In Proceedings of the 18th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Havana, Cuba, 20–23 November 2013.
15. Huang, P.; Huang, Y.; Wang, W.; Wang, L. Deep embedding network for clustering. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014.
16. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
17. Yang, B.; Fu, X.; Sidiropoulos, N.D.; Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017.
18. Yang, X.; Deng, C.; Zheng, F.; Yan, J.; Liu, W. Deep spectral clustering using dual autoencoder network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
19. Wang, J.; Jiang, J. Unsupervised deep clustering via adaptive GMM modeling and optimization. *Neurocomputing* **2021**, *433*, 199–211. [CrossRef]
20. Zhu, D.; Chen, S.; Ma, X.; Du, R. Adaptive Graph Convolution Using Heat Kernel for Attributed Graph Clustering. *Appl. Sci.* **2020**, *10*, 1473. [CrossRef]
21. Krishna, K.; Murty, M.N. Genetic K-means algorithm. *IEEE Trans. Syst. Man Cybern. Part B* **1999**, *29*, 433–439. [CrossRef] [PubMed]
22. Özyer, T.; Liu, Y.; Alhajj, R.; Barker, K. Multi-objective genetic algorithm based clustering approach and its application to gene expression data. In Proceedings of the Third International Conference on Advances in Information Systems, Izmir, Turkey, 20–22 October 2004.
23. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010.
24. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, Paris, France, 18–20 September 2000.
25. Yeung, K.Y.; Ruzzo, W.L. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics* **2001**, *17*, 763–774. [CrossRef] [PubMed]
26. Cai, D.; He, X.; Han, J. Locally consistent concept factorization for document clustering. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 902–913. [CrossRef]
27. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
28. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Statistical Laboratory of the University of California, Berkeley, CA, USA, 21 June–18 July 1965 and 27 December 1965–7 January 1966.