Article

A Genetic XK-Means Algorithm with Empty Cluster Reassignment

Chun Hua ^{1,2}, Feng Li ¹, Chao Zhang ¹, Jie Yang ¹ and Wei Wu ^{1,*}

¹ School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China; chunhua99018074@163.com (C.H.); dscling@126.com (F.L.); chao.zhang@dlut.edu.cn (C.Z.); yangjie@dlut.edu.cn (J.Y.)

² School of Computer Sciences and Technology, Inner Mongolia University for Nationalities, Tongliao 028043, China

* Correspondence: wuweiw@dlut.edu.cn

Received: 11 April 2019; Accepted: 24 May 2019; Published: 2 June 2019

Abstract: K-Means is a well known and widely used classical clustering algorithm. It is easy to fall into local optimum and it is sensitive to the initial choice of cluster centers. XK-Means (eXploratory K-Means) has been introduced in the literature by adding an exploratory disturbance onto the vector of cluster centers, so as to jump out of the local optimum and reduce the sensitivity to the initial centers. However, empty clusters may appear during the iteration of XK-Means, causing damage to the efficiency of the algorithm. The aim of this paper is to introduce an empty-cluster-reassignment technique and use it to modify XK-Means, resulting in an EXK-Means clustering algorithm. Furthermore, we combine the EXK-Means with genetic mechanism to form a genetic XK-Means algorithm with empty-cluster-reassignment, referred to as GEXK-Means clustering algorithm. The convergence of GEXK-Means to the global optimum is theoretically proved. Numerical experiments on a few real world clustering problems are carried out, showing the advantage of EXK-Means over XK-Means, and the advantage of GEXK-Means over EXK-Means, XK-Means, K-Means and GXK-Means (genetic XK-Means).

Keywords: K-Means; genetic mechanism; exploratory disturbance; global convergence; empty-cluster-reassignment

1. Introduction

Clustering algorithms are a class of unsupervised classification methods for a data set (cf. [1–5]). Roughly speaking, a clustering algorithm classifies the vectors in the data set such that distances of the vectors in the same cluster are as small as possible, and the distances of the vectors belonging to different clusters are as large as possible. Therefore, the vectors in the same cluster have the greatest similarity, while the vectors in different clusters have the greatest dissimilarity.

A clustering technique called K-Means is proposed and discussed in [1,2] among many others. Because of its simplicity and fast convergence speed, K-Means is widely used in various research fields. For instance, K-Means is used in [6] for removing the noisy data. A disadvantage of K-Means is that it is easy to fall into local optima. As a remedy, a popular trend is to integrate the genetic algorithm [7,8] with K-means to obtain genetic K-means algorithms [9–23]. K-Means is also combined with fuzzy mechanism to obtain fuzzy C-Means [24,25].

A successful modification of K-Means is proposed in [26], referred to as XK-Means (eXploratory K-Means). It adds an exploratory disturbance onto the vector of the cluster centers so as to jump out of the local optimum and to reduce the sensitivity to the initial centers. However, empty clusters may appear during the iteration of XK-Means, which violates the condition that the number of clusters

should be a pre-given number K and causes damage to the efficiency of the algorithm (see Remark 1 in Section 2.3 below for details). As a remedy, we propose in this paper to modify XK-Means in terms of an empty-cluster-reassignment technique, resulting in an EXK-Means clustering algorithm.

The involvement of the exploratory disturbance in EXK-Means helps to jump out of the local optimum during the iteration. However, in order to guarantee the convergence of the iteration process, the exploratory disturbance has to decrease and tend to zero in the iteration process. Therefore, it is still possible for EXK-Means to fall into local optimum. To further resolve this problem, we follow the aforementioned strategy to combine the genetic mechanism with our EXK-Means, resulting in a clustering algorithm called GEXK-Means.

Numerical experiments on thirteen real world data sets are carried out, showing the higher accuracies of our EXK-Means over XK-Means, and our GEXK-Means over G XK-Means, EXK-Means, XK-Means and K-Means: first, our GEXK-Means achieves the highest S , and the lowest MSE, DB and XB (see the next section for definitions of these evaluation tools) for all of the thirteen data sets. Therefore, GEXK-Means performs better than the other four algorithms. Second, the overall performance of our EXK-Means is a little bit better than that of XK-Means, which shows the benefit of the introduction of our empty cluster reassignment technique.

The numerical experiments also show that the execution times of EXK-Means are a little bit longer than those of K-Means and XK-means, and the execution times of GEXK-Means are the longest in the five algorithms. This is a disadvantage of EXK-Means and GEXK-Means. However, the computer speed is getting high and high in nowadays, and sometimes the computational time does not matter very much in practice if the data set is not very large. In case we do not mind a bit of an increase in the computational time and we care very much about the accuracy, our algorithm may be of value.

A probabilistic convergence of our GEXK-Means to the global optimum is theoretically proved.

This paper is organized as follows. In Section 2, we describe the K-Means, XK-Means, G XK-Means, and our proposed EXK-Means and GEXK-Means. In Section 3, numerical experiments are shown on GEXK-Means and its comparison with K-Means, XK-Means, G XK-Means and EXK-Means. The convergence of GEXK-Means to a globally optimal solution is theoretically proved in Section 4. Some short conclusions are drawn in Section 5.

2. Algorithms

In this section, we first give some notations and describe some evaluation tools. Then, we define the clustering algorithms used in this paper.

2.1. Notations

Let us introduce some notations. Our task is to cluster a set of n genes $\{\mathbf{x}_i, i = 1, 2, \dots, n\}$ into K clusters. Each gene is expressed as a vector of dimension D : $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$. For $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, K$, we define

$$w_{ik} = \begin{cases} 1, & \text{if the } i\text{-th gene belongs to the } k\text{-th cluster,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In addition, we define the label matrix $\mathbf{W} = [w_{ik}]$. We require that each gene belongs to precisely one cluster, and each cluster contains at least one gene. Therefore,

$$\sum_{k=1}^K w_{ik} = 1, \quad i = 1, 2, \dots, n, \quad (2)$$

$$1 \leq \sum_{i=1}^n w_{ik} < n, \quad k = 1, 2, \dots, K. \quad (3)$$

Denote the center of the k -th cluster by $\mathbf{c}_k = (c_{k1}, c_{k2}, \dots, c_{kD})^T$, defined as

$$\mathbf{c}_k = \frac{\sum_{i=1}^n w_{ik} \mathbf{x}_i}{\sum_{i=1}^n w_{ik}}. \quad (4)$$

The Euclidean norm $\|\cdot\|$ will be used in our paper. Then, for any two D -dimensional vectors $\mathbf{y} = (y_1, y_2, \dots, y_D)^T$ and $\mathbf{z} = (z_1, z_2, \dots, z_D)^T$ in R^D , the distance is

$$\|\mathbf{y} - \mathbf{z}\| = \left(\sum_{i=1}^D |y_i - z_i|^2 \right)^{\frac{1}{2}}. \quad (5)$$

2.2. Evaluation Strategies

In our numerical simulation, we will use the following evaluation tools: the mean squared error (MSE), the Xie–Beni index (XB) [12], the Davies–Bouldin index (DB) [13,27], and the separation index (S) [4]. The aim of the clustering algorithms discussed in this paper is to choose the optimal centers $\mathbf{c}_k$'s and the optimal label matrix $\mathbf{W}$ so as to minimize the mean square error MSE. Then, MSE together with the indexes XB, DB and S will be applied to evaluate the outcome of the clustering algorithms.

MSE is defined by

$$MSE = \frac{1}{n} \sum_{k=1}^K \sum_{i=1}^n w_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2. \quad (6)$$

MSE will be used as the evaluation function in the genetic operation of the numerical simulation later on. Generally speaking, lower MSE means better clustering result.

The XB index [12] is defined as follows:

$$XB = \frac{MSE}{d_{min}}, \quad (7)$$

where d_{min} is the shortest distance between cluster centers. Higher d_{min} means better clustering result. As we mentioned above, the MSE is the lower the better. Therefore, lower XB implies better clustering results.

To define the DB index [13,27], we first defined the within-cluster separation S_k as

$$S_k = \left(\frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2 \right)^{\frac{1}{2}}, \quad (8)$$

where C_k (resp. $|C_k|$) denotes the set (resp. the number) of the samples belonging to the cluster k . Next, we define a term R_k for cluster c_k as

$$R_k = \max_{j, j \neq k} \frac{S_k + S_j}{\|\mathbf{c}_k - \mathbf{c}_j\|}. \quad (9)$$

Then, the DB index is defined as

$$DB = \frac{1}{K} \sum_{k=1}^K R_k. \quad (10)$$

Generally speaking, lower DB implies better clustering results.

The separation index S [4] is defined as follows:

$$S = \frac{1}{\sum_{k,j=1; k \neq j}^K |C_k| |C_j|} \sum_{k,j=1; k \neq j}^K |C_k| |C_j| \| \mathbf{c}_k - \mathbf{c}_j \| . \quad (11)$$

Generally speaking, higher S implies better clustering results.

The Nemenyi test [28–30] will be used for evaluating the significance of differences of XK-Means vs. EXK-Means and GXK-Means vs. GEXK-Means, respectively. The function `cdf.chisq` of SPSS software (SPSS Statistics 17.0, IBM, New York, USA) is used to compute the significance probability Pr . The value of Pr is in between 0 and 1. The smaller value of Pr implies the bigger significance of the difference of the two groups. One can say that the difference of the two groups is significant if Pr is less than a particular threshold value. The most often used threshold values are 0.01, 0.05 and 0.1. The threshold value 0.05 will be adopted in this paper.

The relative error $ReError$ defined below will be used for a stop criterion in our numerical iteration process:

$$ReError = \left| \frac{MSE_{t-1} - MSE_t}{MSE_t} \right|, \quad (12)$$

where MSE_t and MSE_{t-1} denote the values of MSE in the current and previous iteration steps, respectively.

2.3. XK-Means

Trying to jump out of the local minimum, the XK-Means algorithm is proposed in [26], where the usual K-Means is modified by adding an exploratory vector onto each cluster center as follows:

$$\mathbf{c}_k^* = \mathbf{c}_k + \theta_k, \quad (13)$$

where θ_k is a D -dimensional exploratory vector at the current step. It is used to disturb the center produced by K-Means operation, and its component is randomly chosen as

$$(\theta_k)_i = \text{rand}(a_i, b_i) * \text{randsign}(i), \quad i = 1, 2, \dots, D, \quad (14)$$

where b_i is a given positive number, and

$$a_i = \beta b_i, \quad (15)$$

with a given factor $\beta \in [0, 1)$. In general, the disturbance should be decreased with the increase of the iteration step. Thus, for a new iteration step, the new value of b_i is set to be

$$b_i^* = \alpha b_i, \quad (16)$$

with a given factor $\alpha \in [0, 1)$.

Remark 1. Empty cluster will not appear in a usual K-Means iteration process. However, it is possible for XK-Means to produce an empty cluster in the iteration process. This happens when the exploratory vector θ_k in Formula (13) drives the center $\mathbf{c}_k$ away from the genes in the k -th cluster, such that all these genes join another cluster in the re-organization stage of the XK-Means and leave the k -th cluster empty. Then, the XK-Means iteration will end up with the number of clusters less than K , which violates the condition that the number of clusters should be K .

2.4. EXK-Means

Due to the disturbance θ_k , the XK-Means algorithm may produce empty clusters during the iteration process, which violates condition (3). The reason for such a cluster to become empty is that it is too close to, and is attracted into other cluster when the centers are disturbed by the θ_k 's. In this

sense, it seems reasonable for such a cluster to “disappear”. However, on the other hand, the empty clusters will damage the clustering efficiency due to the decrease of the number of working clusters.

To resolve this problem, our idea is to re-assign such an empty cluster by a vector that is farthest to its center. Specifically, our EXK-Means modifies the XK-means by applying the following Empty-cluster-reassignment procedure when empty clusters appear after an XK-Means iteration step.

Empty-cluster-reassignment procedure:

1. Let K_0 be the number of empty clusters, $1 \leq K_0 < K$.
2. Find the most marginal point of each non-empty cluster: $\mathbf{x}_k^* = \arg \max_{\mathbf{x}_i \in \mathbf{C}_k} \|\mathbf{x}_i - \mathbf{c}_k\|$, where $\mathbf{C}_k$ is the set of genes in the k -th cluster.
3. Sort $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_K^*\}$ in descending order according to their distances to the corresponding centroids to get $\{\mathbf{x}_1^{**}, \mathbf{x}_2^{**}, \dots, \mathbf{x}_K^{**}\}$.
4. Take the first K_0 genes from $\{\mathbf{x}_1^{**}, \mathbf{x}_2^{**}, \dots, \mathbf{x}_K^{**}\}$ to form K_0 new centers $\{\mathbf{x}_1^{**}\}, \{\mathbf{x}_2^{**}\}, \dots, \{\mathbf{x}_{K_0}^{**}\}$.
5. Adjust the partition of genes according to original centers and the new K_0 centers.

2.5. Genetic Operations

As we argued in the Introduction, although EXK-Means and XK-Means algorithms improve the K-Means on the local minimum issue, but the possibility remains for them to fall into local optimum. We try to combine a genetic mechanism with the EXK-Means to get the global convergence. In particular, we propose to use the following genetic operations:

2.5.1. Label Vectors

For the convenience of genetic operation, in place of the label matrix $\mathbf{W}$, let us introduce the n -dimensional label vector

$$\mathbf{L} = (\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_i, \dots, \mathbf{l}_n)^T, \quad (17)$$

where each component $\mathbf{l}_i \in \{1, 2, \dots, K\}$ represents the cluster label of $\mathbf{x}_i$, as in [10]. Let N denote the population size. Then, we write the population set as $\{\mathbf{L}^j, j = 1, 2, \dots, N\}$.

2.5.2. Initialization

To avoid empty clusters in the initialization stage, we initialize the population as follows. First, the top K components of each $\mathbf{L}^j$ are randomly assigned as a permutation of $\{1, 2, \dots, K\}$. Secondly, the other components of $\mathbf{L}^j$ are assigned as random cluster numbers respectively selected from the uniform distribution of the $\{1, 2, \dots, K\}$.

2.5.3. Selection

The usual roulette strategy is used for the random selection. The probability that an individual $\mathbf{L}^j$ is selected from the existing population to breed the next generation is given by

$$\mathbf{P}(\mathbf{L}^j) = \frac{\mathbf{F}(\mathbf{L}^j)}{\sum_{h=1}^N \mathbf{F}(\mathbf{L}^h)}, \quad j = 1, 2, \dots, N, \quad (18)$$

$$\mathbf{F}(\mathbf{L}^j) = \frac{1}{\frac{1}{n} \sum_{k=1}^K \sum_{i=1}^n w_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|}, \quad (19)$$

where $\mathbf{F}(\mathbf{L}^j)$ is the reciprocal of MSE and represents the fitness value of the individual $\mathbf{L}^j$ in the population.

2.5.4. Mutation

The mutation probability is denoted by P_m , which determines whether an individual L_j will be mutated. If an individual L_j is to be mutated, the translation probability of its component I_i to be k is defined as

$$P_{ik} = P\{I_i = k\} = \frac{2d_{\max}^i - \|x_i - c_k\|}{\sum_{l=1}^K (2d_{\max}^i - \|x_i - c_l\|)}, \quad (20)$$

$$d_{\max}^i = \max_k \{\|x_i - c_k\|\}, \quad (21)$$

where $i = 1, 2, \dots, n$, $k = 1, 2, \dots, K$. To avoid empty individuals after mutation operation, I_i is mutated only when the I_i -th cluster contains more than two genes.

2.5.5. Three Steps EXK-Means

A three-step EXK-Means is applied for rapid convergence. For an individual L , it is updated through the following operations: calculate the cluster centers by using (4) for the given L ; add the exploratory vector and update the cluster centers by using (13); reassign each gene to the cluster with the closest cluster center to form a new individual L ; correct the new L by using the Empty-cluster-reassignment procedure in Section 2.4 if it contains empty cluster(s) at this moment. Repeat the process three times, and finally form an individual L of the next generation.

2.6. Genetic XK-Means (GXK-Means)

The GXK-Means is briefly described as follows:

1. Initialization: Set the population size N , the maximum number of iterations T , the mutation probability P_m , the number of clusters K and the error tolerance E_{Tol} . Let $t = 0$, and choose the initial population $P(0)$ according to Section 2.5.2. In addition, choose the best individual from $P(0)$ and denote it as super individual $L^*(0)$.
2. Selection: Select a new population from $P(t)$ according to Section 2.5.3, and denote it by $P_1(t)$.
3. Mutation: Mutate each individual in $P_1(t)$ according to Section 2.5.4, and get a new population denoted by $P_2(t)$.
4. XK-Means: Perform XK-Means on $P_2(t)$ three times to get the next generation population denoted by $P(t+1)$.
5. Update the super individual: choose the best individual from $P(t+1)$ and compare it with $L^*(t)$ to get $L^*(t+1)$.
6. Stop if either $t = T$ or $ReError \leq E_{Tol}$ (see (12)), otherwise go to 2 with $t \leftarrow t + 1$.

2.7. GEXK-Means (Genetic EXK-Means)

The process of GEXK-Means proposed in this paper is as follows:

1. Initialization: Set the population size N , the maximum number of iterations T , the mutation probability P_m , the number of clusters K and the error tolerance E_{Tol} . Let $t = 0$, and choose the initial population $P(0)$ according to Section 2.5.2. In addition, choose the best individual from $P(0)$ and denote it as super individual $L^*(0)$.
2. Selection: Select a new population from $P(t)$ according to Section 2.5.3, and denote it by $P_1(t)$.
3. Mutation: Mutate each individual in $P_1(t)$ according to Section 2.5.4, and get a new population denoted by $P_2(t)$.
4. EXK-Means: Perform the three steps EXK-Means on $P_2(t)$ according to Section 2.5.5 to get the next generation population denoted by $P(t+1)$.
5. Update the super individual: choose the best individual from $P(t+1)$ and compare it with $L^*(t)$ to get $L^*(t+1)$.
6. Stop if either $t = T$ or $ReError \leq E_{Tol}$ (see (12)), otherwise go to 2 with $t \leftarrow t + 1$.

Let us explain the functions of the four operations in the GEXK-Means: selection, mutation, EXK-Means and updating of the super individual. The selection operation encourages the population to have a good evolution direction. The function of the EXK-Means operation is local search for better individuals. The mutation operation guarantees the ergodicity of the evolution process, which in turn guarantees the appearance of a global optimal individual in the evolution process. Finally, the updating operation of the super individual will catch forever the global optimal individual once it appears.

3. Experimental Evaluation and Results

3.1. Data Sets and Parameters

Thirteen data sets shown in Table 1 are used for evaluating our algorithms. The first five data sets are gene expression data sets, including Sporulation [31], Yeast Cell Cycle [32], Lymphoma [33], and two UCI data sets Yeast and Ecoli. The other eight are UCI data sets, which are not gene express data sets.

As shown in Table 1, Sporulation, Yeast Cell Cycle and Lymphoma data sets contain some sample vectors with missing component values. To rectify these defective data, we follow the strategy adopted in [34–36]: the sample vectors with more than 20% missing components are removed from the data sets. In addition, for the sample vectors with less than 20% missing components, the missing component values are estimated by the KNN algorithm with the parameter $k = 15$ as in [35], where k is the number of the neighboring vectors used to estimate the missing component value (see [34–36] for details). Here, we point out that this parameter k here is different from the index k we have used in this paper for denoting the k -th cluster.

Table 1. Data sets used in experiments.

Data Sets	No. of Vectors n	No. of Vectors with Missing Components <20%	No. of Vectors with Missing Component $\geq 20\%$	No. of Attributes D	No. of Classes K
Sporulation	6023	413	198	7	16
Yeast Cell Cycle	6078	5498	680	77	256
Lymphoma	4022	3166	3	96	150
Yeast	1484	0	0	8	10
Ecoli	336	0	0	8	7
Dermatology	366	8	0	34	6
Glass Identification	214	0	0	10	7
Image Segmentation	2310	0	0	20	7
Wine Quality	4898	0	0	12	7
Wireless Indoor Localization	2000	0	0	7	4
Statlog Vehicle	946	0	0	18	4
Page Blocks Classification	5473	0	0	10	6
Wine	178	0	0	13	3

The values of the parameters used in the computation are set as follows:

Population size $n = 20$	(cf. Sections 2.6 and 2.7)
Mutation probability $P_m = 0.1$	(cf. Sections 2.6 and 2.7)
Error tolerance $E_{Tol} = 0.001$	(cf. Sections 2.2, 2.6 and 2.7)
$\alpha = 0.3$	(cf. (14), (16))
$\beta = 0.95$	(cf. (14), (15))
$T = 150$	(cf. Sections 2.6 and 2.7)

In the experiments, we use two different computers: M1 (Intel (R), Core (TM) i3-8100 CPU and 4 GB RAM, Santa Clara, CA, USA) and M2 (Intel (R), Core (TM) i5-7400 CPU and 8 GB RAM). The software Matlab (Matlab 2017b, Math Works, Natick, MA, USA) is used to implement the clustering algorithms.

3.2. Experimental Results and Discussion

We divide this subsection into three parts. The first part concerns with the performances of the algorithms in terms of MSE, S, DB and XB. The second part demonstrates the significance of differences of the algorithms in terms of Nemenyi Test. The third part presents the computational times of the algorithms. We shall pay our attention mainly on the comparisons of EXK-Means vs. XK-Means and GEXK-Means vs. GXX-Means, respectively, so as to show the benefit of the introduction of our empty-cluster-reassignment technique.

3.2.1. MSE, S, DB and XB Performances

Each of the five algorithms conducted fifty trials on the thirteen data sets. The averages over the fifty trials for the four evaluation criteria (MSE, S, DB and XB) are listed in Tables 2 and 3, devoted to the five gene expression data sets and the other eight UCI data sets, respectively.

Table 2. Average MSE, S, DB and XB on the gene data sets.

Data Sets	Algorithm	MSE (Lower the Better)	S (Higher the Better)	DB (Lower the Better)	XB (Lower the Better)
Yeast Cell Cycle	K-Means	2.3092	3.1637	2.6990	5.0987×10^{-4}
	XK-Means	2.2753	3.1773	2.5595	4.9905×10^{-4}
	EXK-Means	2.2728	3.2170	2.0686	4.7811×10^{-4}
	GXX-Means	2.2623	3.2560	2.4612	3.8156×10^{-4}
	GEXK-Means	2.2572	3.2625	1.8070	3.6357×10^{-4}
Sporulation	K-Means	0.8959	2.7612	1.5781	3.0905×10^{-4}
	XK-Means	0.8968	2.7556	1.5261	3.0761×10^{-4}
	EXK-Means	0.8987	2.7315	1.6226	2.8585×10^{-4}
	GXX-Means	0.8961	2.7510	1.5207	1.7465×10^{-4}
	GEXK-Means	0.8951	2.7674	1.3285	1.5321×10^{-4}
Lymphoma	K-Means	4.8762	7.1725	2.8532	7.4561×10^{-4}
	XK-Means	4.7683	7.2998	2.7294	7.1749×10^{-4}
	EXK-Means	4.7764	7.2890	2.5627	5.7017×10^{-4}
	GXX-Means	4.7520	7.3296	2.3285	5.7929×10^{-4}
	GEXK-Means	4.7244	7.3787	2.1489	5.0597×10^{-4}
Yeast	K-Means	0.1613	0.3106	1.6854	9.6311×10^{-4}
	XK-Means	0.1586	0.3192	1.3592	8.6864×10^{-4}
	EXK-Means	0.1607	0.3234	1.3701	9.0603×10^{-4}
	GXX-Means	0.1581	0.3194	1.2621	6.3508×10^{-4}
	GEXK-Means	0.1560	0.3296	0.9154	4.2514×10^{-4}
Ecoli	K-Means	0.2914	3.3151	1.4452	7.6×10^{-3}
	XK-Means	0.2880	3.2775	1.0973	5.9×10^{-3}
	EXK-Means	0.2382	3.7191	0.7127	3.3×10^{-3}
	GXX-Means	0.2321	3.4022	0.3364	3.1×10^{-3}
	GEXK-Means	0.2268	3.7791	0.3021	1.1×10^{-3}

Table 3. Average MSE, S, DB and XB on the non-gene data sets.

Data Sets	Algorithm	MSE (Lower the Better)	S (Higher the Better)	DB (Lower the Better)	XB (Lower the Better)
Glass Identification	K-Means	1.2886	3.8306	2.0521	8.7987×10^{-3}
	XK-Means	1.1556	4.1060	1.3489	7.7299×10^{-3}
	EXK-Means	1.2113	4.1161	1.4594	8.3280×10^{-3}
	GXK-Means	1.1268	4.1218	0.9689	3.7220×10^{-3}
	GEXK-Means	1.0250	4.1293	0.7230	1.5982×10^{-3}
Image Segmentation	K-Means	63.8058	168.3603	1.4437	3.7125×10^{-4}
	XK-Means	66.8793	169.0878	1.3625	4.7787×10^{-4}
	EXK-Means	59.8254	169.7355	1.2243	4.7210×10^{-4}
	GXK-Means	59.7865	186.6527	1.0697	2.9622×10^{-4}
	GEXK-Means	59.5037	187.7843	1.0269	2.4321×10^{-4}
Page Blocks Classification	K-Means	645.1506	5.6031×10^3	1.3881	6.8264×10^{-4}
	XK-Means	643.3151	5.6031×10^3	1.6223	7.2708×10^{-4}
	EXK-Means	640.8521	5.6301×10^3	1.1316	6.7756×10^{-4}
	GXK-Means	605.8574	5.6896×10^3	0.8693	1.6157×10^{-4}
	GEXK-Means	601.7767	5.6964×10^3	0.7865	1.0283×10^{-4}
Wireless Indoor Localization	K-Mean	10.2066	28.9495	1.6324	4.4231×10^{-4}
	XK-Means	10.1989	28.9495	1.7512	3.4610×10^{-4}
	EXK-Means	10.1962	28.9490	1.14556	5.521×10^{-4}
	GXK-Means	10.1849	28.9210	0.9275	2.3561×10^{-4}
	GEXK-Means	10.0854	28.9840	0.8816	2.2878×10^{-4}
Dermatology	K-Mean	5.7441	20.8749	1.4462	3.3×10^{-3}
	XK-Means	5.8551	20.6550	1.2770	5.7×10^{-3}
	EXK-Means	5.7397	20.7767	1.2425	2.9×10^{-3}
	GXK-Means	5.7420	20.7639	1.2201	2.2×10^{-3}
	GEXK-Means	5.7252	20.9325	0.8816	1.6×10^{-3}
Statlog (Vehicle Silhouettes)	K-Mean	53.8433	271.5335	0.8871	7.005×10^{-4}
	XK-Means	53.8433	271.5335	1.2066	8.8111×10^{-4}
	EXK-Means	53.6535	269.1440	1.0323	6.5618×10^{-4}
	GXK-Means	53.5880	270.6111	0.6289	6.3654×10^{-4}
	GEXK-Means	53.4423	301.5472	0.4893	5.6564×10^{-4}
Wine Quality	K-Mean	14.2767	58.1880	0.9625	2.1481×10^{-4}
	XK-Means	14.2021	58.2228	0.9658	2.1777×10^{-4}
	EXK-Means	14.2090	58.5258	0.9420	2.5431×10^{-4}
	GXK-Means	14.1382	58.5540	0.8499	2.3418×10^{-4}
	GEXK-Means	14.1032	58.6269	0.6973	1.7011×10^{-4}
Wine	K-Means	93.0094	470.2573	0.8236	4.1×10^{-3}
	XK-Means	93.2120	470.2573	0.5436	2.6×10^{-3}
	EXK-Means	93.0092	469.4700	0.6360	2.2×10^{-3}
	GXK-Means	92.9745	470.3015	0.5275	1.8×10^{-3}
	GEXK-Means	92.8682	504.1908	0.3211	1.5×10^{-3}

From Tables 2 and 3, we see that our GEXK-Means achieves the highest S, and the lowest MSE, DB and XB for all the thirteen data sets. Therefore, GEXK-Means performs better than the other four algorithms.

We also observe that the overall performance of our EXK-Means is a bit better than that of XK-Means: EXK-Means is better than XK-Means in terms of all the four clustering criteria (MSE, S, XB and DB) for three of the thirteen data sets; EXK-Means is better in terms of three criteria for three data sets; EXK-Means is better in terms of two criteria for four data sets; and EXK-Means is better in terms of one criteria for three data sets. This means that EXK-Means performs better than XK-Means in nearly two thirds of the cases. (In the total $13 \times 4 = 52$ cases, EXK-Means is better than XK-Means

for $3 \times 4 + 3 \times 3 + 2 \times 4 + 1 \times 3 = 32$ cases.) The better case is marked by black face number in Tables 2 and 3.

To see more clearly the overall performance, in Figures 1–4 for MSE, DB, XB and S evaluations respectively, we further present the average performances of the five algorithms over the thirteen data sets. These figures clearly show that, in the sense of average performance, the proposed GEXK-Means outperforms the other four algorithms, and EXK-Means outperforms K-Means and XK-Means.

Figure 1. Average MSE of thirteen data sets (The value of MSE is the lower the better).

Figure 2. Average DB of thirteen data sets (The value of DB is the lower the better).

Figure 3. Average XB of thirteen data sets (The value of XB is the lower the better).

Figure 4. Average S of thirteen data sets (The value of S is the higher the better).

As an example to show what happens in the iteration processes, a typical iteration process on Yeast Cell Cycle data set is shown in Figures 5–8, presenting the MSE, XB, DB and S curves respectively for the five algorithms.

Figure 5. MSE curves of the algorithms for Yeast Cell Cycle (The value of MSE is the lower the better).

Figure 6. DB curves of the algorithms for Yeast Cell Cycle (The value of DB is the lower the better).

Figure 7. XB curves of the algorithms for Yeast Cell Cycle (The value of XB is the lower the better).

Figure 8. S curves of the algorithms for Yeast Cell Cycle (The value of S is the higher the better).

3.2.2. Nemenyi Test

Table 4 shows the results of Nemenyi Test on MSE, S, DB and XB indexes. We use the threshold value 0.05 for the significance evaluation. For DB index, EXK-Means shows significant difference compared with XK-Means, while EXK-Means does not show significant difference compared with XK-Means for the other three indexes. For all four of the indexes, GEXK-Means shows a significant difference compared with GXK-Means.

Table 4. Nemenyi Test for multiple comparisons.

Groups for Comparison	Evaluation Technique	Pr
XK-Means vs. EXK-Means	MSE	0.8994
	S	0.6691
	DB	0.0421
	XB	0.4539
GXK-Means vs. GEXK-Means	MSE	0.0492
	S	0.0412
	DB	0.0409
	XB	0.0403

3.2.3. Computational Time

Table 5 gives the average computational times over the fifty runs for each data set. It shows that the computational times of EXK-Means are a little bit longer than those of K-Means and XK-means, and the computational times of GEXK-Means are a little bit longer than those of GXK-Means. This indicates that the introduction of our empty-cluster-reassignment technique increases the computational time. However, our algorithms are better if we do not mind a bit of increase in the computational time and we care very much about the accuracy.

Table 5. Average running times (in seconds).

Data Sets	Evaluation Technique	K-Means	XK-Means	EXK-Means	GXK-Means	GEXK-Means	Machine
Sporulation	MSE	11.368	11.689	12.56	289.656	688.552	M2
	S	11.464	11.792	12.61	256.782	689.457	M2
	DB	10.424	10.569	10.480	239.529	662.183	M2
	XB	9.632	9.881	9.878	269.364	396.33	M2
Yeast Cell Cycle	MSE	58.65	60.213	64.52	4867.560	5432.23	M2
	S	58.87	60.351	65.11	4789.108	5433.69	M2
	DB	69.425	72.37	73.06	4965.682	5654.75	M2
	XB	52.584	56.321	59.66	4754.538	5014.56	M2
Lymphoma	MSE	69.425	70.43	74.332	4962.224	5321.6	M2
	S	69.661	71.34	74.994	4987.300	5324.1	M2
	DB	72.135	77.483	74.286	4753.626	5332.42	M2
	XB	62.526	66.36	68.665	4665.186	5026.559	M2
Glass Identification	MSE	0.3675	0.368	0.2975	19.566	20.824	M1
	S	0.3678	0.371	0.2980	19.960	20.885	M1
	DB	0.505	0.535	0.536	19.325	27.687	M1
	XB	0.267	0.303	0.304	16.263	19.806	M1
Image Segmentation	MSE	4.552	4.230	4.336	298.128	361.755	M1
	S	4.578	4.257	4.402	296.867	363.455	M1
	DB	4.564	4.663	4.718	286.692	382.924	M1
	XB	4.565	4.131	4.183	263.960	332.092	M1

Table 5. Cont.

Data Sets	Evaluation Technique	K -Means	XK -Means	EXK -Means	GXK -Means	GEXK -Means	Machine
Page Blocks Classification	MSE	4.039	4.078	4.122	382.663	395.273	M1
	S	4.165	4.216	4.269	378.632	397.421	M1
	DB	7.859	7.649	7.947	376.657	520.118	M1
	XB	4.596	4.622	4.264	296.186	390.57	M1
Yeast	MSE	1.810	1.964	1.870	159.200	165.08	M1
	S	1.914	1.998	1.978	161.230	167.1	M1
	DB	3.630	3.784	3.800	163.620	241.346	M1
	XB	1.718	1.817	1.847	148.960	161.346	M1
Wireless Indoor Localization	MSE	1.428	1.430	1.433	119.630	122.278	M1
	S	1.473	1.459	1.62	120.775	124.512	M1
	DB	2.843	2.760	2.840	124.360	176.689	M1
	XB	1.436	1.445	1.418	98.641	120.628	M1
Ecoli	MSE	0.408	0.437	0.437	29.611	30.494	M1
	S	0.409	0.441	0.439	29.845	30.569	M1
	DB	0.736	0.858	0.882	28.010	43.321	M1
	XB	0.399	0.447	0.490	29.380	31.22	M1
Dermatology	MSE	0.634	0.635	0.710	23.450	24.829	M1
	S	0.639	0.651	0.786	24.687	25.854	M1
	DB	0.78	0.832	0.857	24.312	26.758	M1
	XB	0.406	0.542	0.516	23.720	26.811	M1
Statlog (Vehicle Silhouettes)	MSE	0.705	0.726	0.768	53.856	57.061	M1
	S	0.712	0.732	0.798	55.289	58.067	M1
	DB	1.298	1.303	1.351	58.450	77.966	M1
	XB	0.590	0.634	0.642	43.892	51.137	M1
Wine Quality	MSE	4.538	4.658	4.758	386.680	444.963	M1
	S	4.563	4.713	4.799	379.668	446.921	M1
	DB	8.435	8.512	8.571	412.654	565.498	M1
	XB	4.463	4.499	4.635	356.215	421.432	M1
Wine	MSE	0.159	0.182	0.179	10.226	11.133	M1
	S	0.161	0.188	0.182	10.129	11.186	M1
	DB	0.267	0.304	0.309	11.636	13.852	M1
	XB	0.152	0.172	0.174	8.385	10.130	M1

4. Convergence

In this section, the convergence properties of GEXK-Means are analyzed. It is clear that there exist $m = K^n$ possible solutions when classifying n genes into K clusters. As mentioned in Section 2.5, every possible solution can be denoted as a label vector $\mathbf{L}$. Therefore, the number of all possible individuals is m . Let $\mathcal{L}^*$ be the set of global optimal individuals with maximum fitness value.

For an individual $\mathbf{L}$ that will to be mutated, according to Equation (20), we have

$$\mathbf{P}_{ik} = \mathbf{P}\{\mathbf{l}_i = k\} = \frac{2d_{\max}^i - \|x_i - \mathbf{c}_k\|}{\sum_{l=1}^K (2d_{\max}^i - \|x_i - \mathbf{c}_l\|)},$$

where $i = 1, 2, \dots, n$, $k = 1, 2, \dots, K$, and $d_{\max}^i > \|x_i - \mathbf{c}_k\| > 0$. Therefore, $2d_{\max}^i - \|x_i - \mathbf{c}_k\| > 0$, and $\mathbf{P}_{ik} > 0$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, K$. We note that the number of genes and the number of clusters are finite. Therefore, $\mathbf{P}_{ik}$ ($i = 1, 2, \dots, n$, $k = 1, 2, \dots, K$) has lower bound denoted by $M > 0$. This means that every gene can be mutated into any one cluster with positive probability. In particular, $\mathbf{L}$ can be mutated into any other individuals with positive probability. Recall that

$\mathbf{P}(t) = \{\mathbf{L}^1(t), \mathbf{L}^2(t), \dots, \mathbf{L}^N(t)\}$ is the population at step t . Let $\mathbf{P}_{\mathbf{L}^j(t) \rightarrow \mathcal{L}^*}$ stand for the probability in which $\mathbf{L}^j(t)$ is mutated to one of the global optimal individuals. Then

$$\mathbf{P}_{\mathbf{L}^j(t) \rightarrow \mathcal{L}^*} > M^n, j = 1, 2, \dots, N. \quad (22)$$

Let $\mathbf{P}_{Mutation}$ stand for the probability generating the optimal individual in $\mathbf{P}_2(t)$ by mutation operation. Then,

$$\begin{aligned} \mathbf{P}_{Mutation} &= \sum_{j=1}^N P_m \mathbf{P}(\mathbf{L}^j(t)) (\mathbf{P}_{\mathbf{L}^j(t) \rightarrow \mathcal{L}^*}) \\ &> P_m \sum_{j=1}^N \mathbf{P}(\mathbf{L}^j(t)) M^n \\ &= P_m M^n \\ &> 0, \end{aligned} \quad (23)$$

where $P_m > 0$ is the mutation probability, $\mathbf{P}(\mathbf{L}^j(t)) > 0$ is the selection probability defined by Equation (18), and $\sum_{j=1}^N \mathbf{P}(\mathbf{L}^j(t)) = 1$.

Theorem 1. When the GEXK-Means defined in Section 2.7 is applied for the classification of a given data set, the global optimal classification result for the data set will appear and will be caught with probability 1 in an infinite evolution iteration process of the GEXK-Means.

Proof. Along with the evolution process, the updating operation of the super individual will keep the super individual denoted by $\mathbf{L}^*(t)$ of every generation $t = 0, 1, 2, \dots$. According to (22), we know that the $\mathbf{L}^*(t)$ may become a global optimal individual with positive probability. According to the Small Probability Event Principle [37,38], the global optimum individual will appear in the super individual sequence with probability 1 when the evolution iteration process goes to infinity. This proves the global convergence of GEXK-Means. $\square$

We remark that the global convergence stated above is of a theoretical and probabilistic nature. It does not guarantee that the convergence to a global optimum can be reached in finite number of GEXK-Means iterations.

5. Conclusions

XK-Means (eXploratory K-Means) is a popular data clustering algorithm. However, empty clusters may appear during the iteration of XK-Means, which violates the condition that the number of clusters should be K and causes damage to the efficiency of the algorithm. As a remedy, we define an empty-cluster-reassignment technique to modify XK-Means when empty clusters appear, resulting in an EXK-Means clustering algorithm. Furthermore, we combine the EXK-Means with genetic mechanism to form a GEXK-Means clustering algorithm.

Numerical simulations are carried out on the comparison of K-Means, XK-Means, EXK-Means and GXK-Means (genetic XK-Means) and GEXK-Means. The evaluation tools include the mean squared error (MSE), the Xie–Beni index (XB), the Davies–Bouldin index (DB) and the separation index (S). The Nemenyi Test for multiple comparisons is also done on MSE, S, DB and XB, respectively. Thirteen real world data sets are used for the simulation. The running times of these algorithms are also considered.

The conclusions we draw from the simulation results are as follows: first, the overall performances of EXK-Means in terms of the four indexes outperform those of XK-Means, and the overall performances of GEXK-Means outperform those of GXK-Means. This shows the effectiveness of the introduction of the empty-cluster-reassignment technique. Secondly, if we take the threshold

value as 0.05 for the Nemenyi Test, then GEXK-Means shows a significant difference compared with GXX-Means for all four of the indexes. However, EXK-Means shows a significant difference compared with XK-Means only for the DB index. Thirdly, our EXK-Means and GEXK-Means take a little bit more computational time than XK-Means and GXX-Means, respectively.

The following global convergence of the GEXK-Means is also theoretically proved: the global optimum will appear and will be caught in the evolution process of GEXK-Means with probability 1.

Author Contributions: C.H. developed the mathematical model, carried out the numerical simulations and wrote the manuscript; W.W. advised on developing the learning algorithms and supervised the work; F.L. contributed to the theoretical analysis; C.Z. and J.Y. helped in the numerical simulations.

Funding: This research was supported by the National Science Foundation of China (No: 61473059), and the Fundamental Research Funds for the Central Universities of China.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Steinhaus, H. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci.* **1956**, *3*, 801–804.
2. Macqueen, J. Some Methods for Classification and Analysis of MultiVariate Observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Los Angeles, CA, USA, 21 June–18 July 1967.
3. Wlodarczyk-Sielicka, M. Importance of Neighborhood Parameters During Clustering of Bathymetric Data Using Neural Network. In Proceedings of the 22nd International Conference, Duruskinkai, Lithuania, 13–15 October 2016.
4. Du, Z.; Wang, Y. PK-Means: A New Algorithm for Gene Clustering. *Comput. Biol. Chem.* **2008**, *32*, 243–247. [[CrossRef](#)] [[PubMed](#)]
5. Lin, F.; Du, Z. A Novel Parallelization Approach for Hierarchical Clustering. *Parallel Comput.* **2005**, *31*, 523–527.
6. Santhanam, T.; Padmavathi, M.S. Application of K-Means and Genetic Algorithms for Dimension Reduction by Integrating SVM for Diabetes Diagnosis. *Procedia Comput. Sci.* **2015**, *47*, 76–83. [[CrossRef](#)]
7. Deep, K.; Thakur, M. A New Mutation Operator for Real Coded Genetic Algorithms. *Appl. Math. Comput.* **2007**, *193*, 211–230. [[CrossRef](#)]
8. Ming, L.; Wang, Y. On Convergence Rate of a Class of Genetic Algorithms. In Proceedings of the World Automation Congress, Budapest, Hungary, 24–26 July 2006.
9. Maulik, U. Genetic Algorithm Based Clustering Technique. *Pattern Recognit.* **2000**, *33*, 1455–1465. [[CrossRef](#)]
10. Jones, D.R.; Beltramo, M.A. Solving Partitioning Problems with Genetic Algorithms. In Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, USA, 13–16 July 1991.
11. Zheng, Y.; Jia, L.; Cao, H. Multi-Objective Gene Expression Programming for Clustering. *Inf. Technol. Control* **2012**, *41*, 283–294. [[CrossRef](#)]
12. Xie, X.L.; Beni, G. A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 841–847. [[CrossRef](#)]
13. Liu, Y.G. Automatic Clustering Using Genetic Algorithms. *Appl. Math. Comput.* **2011**, *218*, 1267–1279. [[CrossRef](#)]
14. Krishna, K.; Murty, M.N. Genetic K-Means Algorithm. *IEEE Trans. Syst. Man Cybern.* **1999**, *29*, 433–439. [[CrossRef](#)]
15. Bouhmala, N.; Viken, A. Enhanced Genetic Algorithm with K-Means for the Clustering Problem. *Int. J. Model. Optim.* **2015**, *5*, 150–154. [[CrossRef](#)]
16. Sheng, W.G.; Tucker, A. Clustering with Niching Genetic K-means algorithm. In Proceedings of the 6th Annual Genetic and Evolutionary Computation Conference (GECCO 2004), Seattle, WA, USA, 26–30 June 2004.
17. Zhou, X.B.; Gu, J.G. An Automatic K-Means Clustering Algorithm of GPS Data Combining a Novel Niche Genetic Algorithm with Noise and Density. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 392–397. [[CrossRef](#)]

18. Islam, M.Z.; Estivill-Castro, V.; Rahman, M.A.; Bossomaier, T. Combining K-Means and a Genetic Algorithm through a Novel Arrangement of Genetic Operators for High Quality Clustering. *Expert Syst. Appl.* **2018**, *91*, 402–417. [\[CrossRef\]](#)
19. Michael, L.; Sumitra, M. A Genetic Algorithm that Exchanges Neighboring Centers for k-means clustering. *Pattern Recognit. Lett.* **2007**, *28*, 2359–2366.
20. Ishibuchi, H.; Yamamoto, T. Fuzzy Rule Selection by Multi-objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining. *Fuzzy Sets Syst.* **2004**, *141*, 59–88. [\[CrossRef\]](#)
21. Zubova, J.; Kurasova, O. Dimensionality Reduction Methods: The Comparison of Speed and Accuracy. *Inf. Technol. Control* **2018**, *47*, 151–160. [\[CrossRef\]](#)
22. Wozniak, M.; Polap, D. Object Detection and Recognition via Clustered Features. *Neurocomputing* **2018**, *320*, 76–84. [\[CrossRef\]](#)
23. Anusha, M.; Sathiaselvan, G.R. Feature Selection Using K-Means Genetic Algorithm for Multi-objective Optimization. *Procedia Comput. Sci.* **2015**, *57*, 1074–1080. [\[CrossRef\]](#)
24. Bezdek, J.C.; Ehrlich, R. FCM: The Fuzzy C-Means Clustering Algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [\[CrossRef\]](#)
25. Indrajit, S.; Ujjwal, M. A New Multi-objective Technique for Differential Fuzzy Clustering. *Appl. Soft Comput.* **2011**, *11*, 2765–2776.
26. Lam, Y.K.; Tsang, P.W.M. eXploratory K-Means: A New Simple and Efficient Algorithm for Gene Clustering. *Appl. Soft. Comput.* **2012**, *12*, 1149–1157. [\[CrossRef\]](#)
27. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [\[CrossRef\]](#)
28. Liu, Y.W.; Chen, W.H. A SAS Macro for Testing Differences among Three or More Independent Groups Using Kruskal-Wallis and Nemenyi Tests. *J. Huazhong Univ. Sci. Tech.-Med.* **2012**, *32*, 130–134. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Nemenyi, P. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
30. Fan, Y.; Hao, Z.O. *Applied Statistics Analysis Using SPSS*, 1st ed.; China Water Conservancy and Hydroelectricity Publishing House: Beijing, China, 2003; pp. 138–152.
31. Chu, S.; DeRisi, J. The Transcriptional Program of Sporulation in Budding Yeast. *Science* **1998**, *282*, 699–705. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Spellman, P.T. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces Cerevisiae* by Microarray Hybridization. *Mol. Biol.* **1998**, *9*, 3273–3297. [\[CrossRef\]](#)
33. Alizadeh, A.A.; Eisen, M.B. Distinct Types of Diffuse Large B-cell Lymphoma Identified by Gene Expression Profiling. *Nature* **2000**, *403*, 503–511. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Yoon, D.; Lee, E.K. Robust Imputation Method for Missing Values in Microarray Data. *BMC Bioinform.* **2007**, *8*, 6–12. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Troyanskaya, O.; Cantor, M. Missing Value Estimation Methods for DNA Microarrays. *Bioinformatics* **2001**, *17*, 520–525. [\[CrossRef\]](#)
36. Corso, D.E.; Cerquitelli, T. METATECH: Meteorological Data Analysis for Thermal Energy CHaracterization by Means of Self-Learning Transparent Models. *Energies* **2018**, *11*, 1336. [\[CrossRef\]](#)
37. Liu, G.G.; Zhuang, Z.; Guo, W.Z. A novel particle swarm optimizer with multi-stage transformation and genetic operation for VLSI routing. *Energies* **2018**, *11*, 1336.
38. Rudolph, G. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Trans. Neural Netw.* **1994**, *5*, 96–101. [\[CrossRef\]](#) [\[PubMed\]](#)