A Parallel Gene Expression Clustering Algorithm Based on Producer-Consumer Model

Lei Yang[1(✉)], Xin Hu[1], Kangshun Li[1], Wensheng Zhang[2], Yaolang Kong[1], Rui Xu[1], and Dongya Wang[3]

[1] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
yanglei_s@scau.edu.cn

[2] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[3] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, UK

Abstract. Clustering is one of the important tasks of machine learning. Gene Expression Programming (GEP) is used to solve clustering problems because of its strong global searching ability. In order to solve the limitation of lower rate of convergence and easy falling into optimal local solution in the traditional GEP clustering process, this paper proposes a parallel GEP clustering algorithm based on the producer-consumer model (PGEPC/PCM), which parallelizes the time-consuming operations such as fitness calculation, recombination, and mutation in GEP clustering analysis to speed up, improves the calculation method of fitness function to enable it to cluster automatically. This algorithm can fast calculate accurate clustering center points in parallel. Extensive experiments on four widely used benchmark Iris, Wine, Soybean and Seeds from the UCI machine learning data sets are conducted to investigate the influence of algorithmic component and results are compared with traditional GEP clustering algorithm. These comparisons demonstrate the competitive efficiency of the proposed algorithm.

Keywords: Gene Expression Programming · Clustering algorithm · Producer-consumer model · Clustering center

1 Introduction

Data mining and knowledge discovery have become an important research topic with the rapid expansion of data. Clustering is an important task of data mining. Because the accurate clustering requires corresponding algorithms, it is particularly important to design an efficient and precise algorithm. The design of the clustering algorithm depends on the type of data and the purpose of cluster analysis. The traditional clustering methods have made related researches in clustering techniques from different perspectives, but many existing algorithms

K. Li et al. (Eds.): ISICA 2019, CCIS 1205, pp. 97–111, 2020.
https://doi.org/10.1007/978-981-15-5577-0_8

have difficulty in automatically clustering data without prior knowledge of data. Evolutionary algorithms are applied to cluster analysis because of their high degree of parallelism, randomization, adaptive search, etc. Murthy and Chowdhury [13] proposed a clustering algorithm based on genetic algorithm (GA) that uses binary coding. Each data unit occupies one bit of an individual chromosome, due to the chromosome length limitation, it can only solve the clustering problem of small data sets. Bandyopadhyay and Maulik [2] proposed an improved GA clustering algorithm that uses cluster center points instead of individual coding schemes in chromosomes, which enormously reduced the chromosome length and can handle larger data sets. Yu Chen, Changjie Tang et al. [3] proposed an auto-clustering algorithm based on gene expression programming. Daihong Jiang and Sanyou Zhang et al. [11] combine the advantages of two kinds of algorithms. From the analysis of gene expression programming and K-Means algorithm, which realized the algebraic operation of gene clustering under the condition of unknown cluster division information, ensuring all grammar of chromosome evolution through this algorithm is correct. It not only avoids the large use of computational resources for editing illegal chromosomes, but also allows the use of chromosomes for modification, the experimental results are better than the traditional K-means clustering algorithm. Yifei Zheng, Lixin Jia, Hui Cao [15] proposed a multi-objective gene expression clustering algorithm, which can automatically determine the number of data sets and appropriate partitions. Hongguo Cai, Changan Yuan [8] proposed a serial clustering algorithm based on gene expression programming, which makes use of the parallelism advantage of Gene Expression Programming (GEP) to combine with the existing serial clustering DBSCAN algorithm. It makes the following program parallelized and improves the efficiency of the algorithm. Clustering algorithms based on gene expression programming have achieved some results. However, these algorithms also have some shortcomings, such as the low speed of operating, the requirement of prior knowledge when clustering, and falling into local optimum when it is disturbed by noises. Therefore, how to solve the problems existing in the traditional GEP algorithm in cluster analysis and obtain more accurate results will be the focus of this paper.

Aiming at the slow running speed and instability of the traditional gene expression programming, and also it is easy to fall into local optimal. This paper proposes a parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM). This algorithm adopts the producer-consumer model to parallelize. It performs multi-thread parallelization on the calculation of fitness, selection, recombination, mutation, etc. in GEP clustering analysis to increase the speed and improves the calculation method of fitness function to enable it to cluster automatically. In this paper, Extensive experiments carried out to compare the average accuracy, the highest accuracy, the running time and other related indicators of the parallel gene expression clustering algorithm based on producer-consumer model and traditional gene expression programming clustering algorithm in Iris, Wine, Soybean, and Seeds from UCI public data sets. The results show that the parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) has a significant improvement in the efficiency of the algorithm.

The remainder of this study is organized as follows. Section 2 presents a discussion of the traditional Clustering analysis based on GEP. Section 3 presents our new parallel gene expression clustering algorithm based on producer-consumer model. Section 4 presents the experimental results and discussion. Finally, Sect. 5 concludes this study and presents future research directions.

2 Clustering Algorithm Based on Gene Expression Programming

2.1 Gene Expression Programming

The gene expression programming proposed by Ferreira [6] is an extension of Genetic Programming (GP), which combines the simple and fast characteristics of Genetic Algorithm's fixed-length linear coding and the flexible and diverse advantages of Genetic programming tree structure. It can solve complex problems with simple coding. The concept of functions in GEP is quite extensive, it includes the intermediate structure of any other non-terminal in the system, the set of functions can include the arithmetic symbols of the problem domain related to the application. Gene expression programming is widely used in symbol regression [16,17], classification [10,14], clustering [12], forecast [5,7,9], combinatorial optimization [1], modeling [4] and resource management [18] because of the fast random searching ability, potential parallelism, scalability, and robustness.

Gene expression programming uses a particular description method different from the genetic algorithm, which primarily uses a generalized hierarchical computer program to describe the problem. The formal description of this generalized hierarchical computer program requires two types of symbols, the terminators and the functions, which are meta-languages for constructing a program in gene expression programming. The chromosome of GEP consists of one or more genes through junction function and terminals, each gene consists of head, and tail, the head of a gene consists of a set of terminals and a set of functions, the tail of the gene only consists of the set of terminals. For each problem, the length of the head is selected in advance, and the length t of the tail is a function of the head lengths h and n, where n is the number of arguments to the function with the most variables, t is obtained by Eq. 1 below.

$$t = h(n-1) + 1 \tag{1}$$

A gene is transformed into an expression by parsing, Moreover, the functions and terminals of the gene are filled from top to bottom in a hierarchical traversal. The basic GEP genetic operators include mutations, insertion sequences, and recombination. Mutation can introduce new nodes into genes, which is the most efficient operator among all the operators with modification ability. To ensure the gene can be expressed generally after mutation. The category of nodes would be different, according to the place where mutation happened. When a mutation occurs in the head of the gene, the node after mutation can be a function or

terminal. If a mutation happened in the tail, the node only could be the terminal. Mutations can take place anywhere in the gene, without regard to the type of the original type. Some of the nodes at the position of mutation, which can be modified randomly based on the above rules. The insertion sequence elements of GEP are fragments of the gene locus that can be activated and jump to another location of the chromosome. There are three kinds of transposable elements in GEP: (1) Short fragments whose the beginning position is function or terminal are transposed to the head of genes except for the root (insertion sequence elements or IS elements); (2) Short fragments with a function at the first position are transposed to the root of genes (root IS elements or RIS elements); and (3) entire genes are transposed into the beginning of chromosomes. There are three recombinations in GEP: single-point recombination, two-point recombination, and genetic recombination. In all cases, two randomly selected parent chromosomes pair and interacted with each other. Gene recombination produces different arrangements of existing genes. The evolutionary ability of GEP is not only based on gene rearrangement, but also based on the continuous generation of new genetic material, which is caused by mutation and insertion sequence.

2.2 Cluster Analysis Based on Gene Expression Programming

Cluster analysis is the process of grouping a collection of physical or abstract objects into multiple classes of similar objects. The goal of cluster analysis is to measure the similarities between different data sources and to classify data sources into different clusters. Clustering can be used as an independent tool to obtain the distribution of data, which also can observe the characteristics of each cluster of data, and focus on further analysis of specific clusters. Cluster analysis can also be used as a preprocessing step for other algorithms. The clustering algorithm based on gene expression programming does not need any prior knowledge of the data sets, which can automatically divide clusters and complete cluster analysis.

Clustering Algorithm Based on GEP

(1) Encoding. Due to the particularity of the clustering problem, the encoding mode uses the single-gen encoding, which consists of a head and a tail. The range of the header encoding is || or &&, and the tail encoding is an individual instance. According to the uniform distribution, the data sequence numbers are randomly extracted from the data set to form a tail sequence. The length of the head is n, and the length of the tail is $n + 1$.

(2) Cluster Fitness Function. The fitness function is the driving force of GEP population evolution, which can make the algorithm evolve in the required direction. In most cases, an individual's fitness evaluation consumes most of the GEP running time. Choosing different fitness functions will affect the quality of the

evolutionary results directly, if the fitness function is selected improperly, it will make a consequence that the iteration does not converge or converge to an unrelated solution. The GEP clustering analysis draws on the principle of "survival of the fittest" in nature to give the fitness function of the corresponding gene expression programming. In other words, the closer the genetic expression is to the actual observation, the higher the fitness will be. The fitness function in the GEP clustering algorithm is defined as follows in Eqs. 2 and 3.

$$f = \frac{1}{1+E} \tag{2}$$

$$E = \sum_{i=1}^{k} \sum_{p \in \mathcal{C}_i} |p - m_i|^2 \qquad m_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j \tag{3}$$

In this formula, the k is the number of clusters in the data set, and the p is the point in space. n_i is the number of data points in the cluster C_i. m_i is the average of cluster C_i. It is calculated by taking the arithmetic mean of the dimensions of all the elements in the cluster. After decoding the chromosomal information, the coordinates of the center points of each possible cluster are obtained, and then the data points of the data set are sequentially assigned to the nearest cluster according to the Euclidean distance from each cluster center. Recalculate the center coordinates of each cluster after clustering, and calculate the sum of squared errors E of all data in the data set. The smaller the E value is, the smaller the cluster will be as compact and independent as possible.

(3) Procedure of GEP Clustering Algorithm. The procedure of the clustering algorithm based on GEP is shown below. Firstly, the data is normalized and preprocessed when the initial training set data is imported. Then, the GEP Algorithm is used to cluster the training set data and get the cluster center point. Then determine whether the fitness of the best individual is ranked in the top ten of the historical operation record. If so, put the resulting individuals in this calculation into the best individuals to compile statistics. Otherwise, discard it. Finally, this algorithm integrates the best individuals in the statistical fitness set to derive rules.

3 Parallel Gene Expression Programming Clustering Algorithm Based on Population Migration Strategy (PGEPC/PCM)

In order to study the shortcomings of the basic gene expression programming clustering algorithm and the necessity of improving the GEP clustering algorithm, in this section, we applied the basic gene expression clustering algorithm to four common data sets, study the performance of the basic gene expression clustering algorithm on the four public data sets and analyze its inadequacies. After that, the parallel gene expression clustering algorithm based on producer-consumer model was proposed.

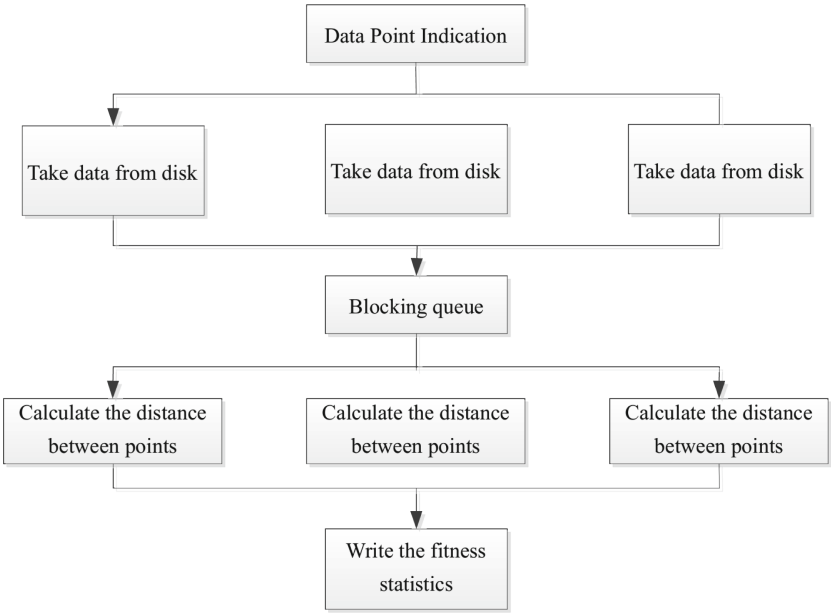

Fig. 1. Fitness calculation parallelization process

3.1 Inadequacies of Basic Gene Expression Programming Clustering Algorithm

Speed of the Algorithm. The main steps of GEP clustering operation are calculating fitness values, recombination, mutation, Furthermore, loop the steps above. The individual fitness calculation process is as follows, for each discrete point, the nearest center point is selected as the cluster to which it is assigned, each point does not need to rely on the calculation results of other points to calculate the center point of the cluster to which it will be assigned. The algorithm costs the most time while calculating the fitness. And it is the main module that limits the efficiency of the program. It needs to adjust the method of individual fitness calculation to improve the efficiency of the algorithm.

Automatic Clustering. After the GEP completes the clustering of each data point, the algorithm determines whether there are clusters that can merge in the cluster set and merges them. Head coding does not have to be involved in recombination and mutation operations, the traditional individual fitness function does not perform automatic clustering well.

3.2 Parallel GEP Clustering Algorithm Based on Producer-Consumer Model (PGEPC/PCM)

The Clustering Algorithm with Parallel Gene Expression Programming based on producer-consumer model proposed in this paper will optimize the basic GEP clustering algorithm from two aspects: The first is parallelizing the time-consuming operations in the algorithm to promote the speed of calculation; The second is improving the fitness function to enable automatic clustering.

Algorithm Local Parallelization

Parallel Method Mining. The calculation of fitness value is the main factor limiting the efficiency of GEP clustering algorithm, Parallel analysis of the phase is needed to improve the efficiency of the algorithm. Design to create a separate thread for each point to calculate the point to which it will be assigned. However, in the case of an enormous amount of data, assuming that there are thousands of pieces of data, it is necessary to open another thread to calculate. Therefore, the system resources will be consumed tremendously.

To solve this problem, we must analyze the critical resources, with the sum of discrete points and distances firstly. Assuming that there are n workers (threads) randomly selecting a point from thousands of points to classify, and then gather statistics of the sum of the distances within the classes. We can use the producer-consumer model to realize the algorithm parallelism. The producer-consumer model is significant in the operating system, which describes a mechanism of waiting and notifying. The producer-consumer is a classic problem in the thread model: producers and consumers share the same storage space during the same period. The producer is responsible for taking out data and stuffing it into the blocking queue. The consumer is responsible for taking out the statistical fitness results from the queue. This parallel computing model is adopted to reduce system overhead and improve system operation efficiency. The algorithm flow is shown in Fig. 1.

This algorithm's description can be divided into the following steps: Randomly take a data point and get the data from the disk with the label; Put the extracted point data into the blocking queue; Calculate the point of the thread taken from the blocking queue and calculate its distance from the center cluster point; Write the calculated data to statistical fitness.

BWP Fitness

Definition 1. *Definition of baw(j,i): Let* $K = \{X, R\}$ *be the cluster space, where* $X = \{x_1, x_2, \ldots, x_n\}$ *. Assuming that n sample objects are clustered into class c, the cluster distance baw(j, i) of the i-th sample defining the j-th class is the sum of the minimum inter-class distance and the intra-class distance of the sample. That is:*

$$baw(j,i) = b(j,i) + w(j,i) = \min_{1\le k\le c, k\neq j}\left(\frac{1}{n_k}\sum_{p=1}^{n_k}\left\|x_p^k - x_i^j\right\|^2\right) + \frac{1}{n_j - 1}\sum_{q=1,q\neq i}^{n_j}\left\|x_q^j - x_i^j\right\|^2 \tag{4}$$

Definition 2. *Definition of bsw(j,i): Let $K = \{X, R\}$ be the cluster space, where $X = \{x_1, x_2, \ldots, x_n\}$. Assuming that n sample objects are clustered into class c, the cluster separation distance bsw(j, i) of the i-th sample defining the j-th class is the difference between the minimum inter-class distance and the intra-class distance of the sample. That is:*

$$baw(j,i) = b(j,i) - w(j,i) = \min_{1\le k\le c, k\neq j}\left(\frac{1}{n_k}\sum_{p=1}^{n_k}\left\|x_p^k - x_i^j\right\|^2\right) - \frac{1}{n_j - 1}\sum_{q=1,q\neq i}^{n_j}\left\|x_q^j - x_i^j\right\|^2 \tag{5}$$

According to $baw(j,i)$ and $bsw(j,i)$, Let $K = X, R$ be the cluster space, where $X = \{x_1, x_2, \ldots, x_n\}$. Assuming that n sample objects are clustered into class c, defining between-withing proportion (BWP) of the i-th sample of class j. The index $BWP(j,i)$ is the ratio of the clustering dispersion distance and the clustering distance of the sample. That is:

$$BWP(j,i) = \frac{bsw(j,i)}{baw(j,i)} = \frac{b(j,i) - w(j,i)}{b(j,i) + w(j,i)} = \frac{\min_{1\le k\le c, k\neq j}\left(\frac{1}{n_k}\sum_{p=1}^{n_k}\left\|x_p^{(k)} - x_i^{(j)}\right\|^2\right) - \frac{1}{n_j-1}\sum_{q=1,q\neq i}^{n_j}\left\|x_q^{(j)} - x_i^{(j)}\right\|^2}{\min_{1\le k\le c, k\neq j}\left(\frac{1}{n_k}\sum_{p=1}^{n_k}\left\|x_p^{(k)} - x_i^{(j)}\right\|^2\right) + \frac{1}{n_j-1}\sum_{q=1,q\neq i}^{n_j}\left\|x_q^{(j)} - x_i^{(j)}\right\|^2} \tag{6}$$

When the BWP fitness calculation is running, the distance to each center point needs to be calculated for each point, and the shortest distance to the points in the other clusters is calculated after the classification. The calculation amount is several times larger than the previous calculation of the distance within the group. When the operating environment is officially running the algorithm, it takes a high time cost, and eventually the data volume is too large to run. However, the main time-consuming part of the program operation is the calculation of the distance between groups. When transplanting into GEP clustering, a compromise method is adopted, and it is not necessary to accurately calculate the distance between groups, and the main idea is extracted to perform an approximate calculation.

Because the GEP chromosomes are limited in length, they only contain a limited number of points as cluster centers. It is not necessary to cluster all the points and then calculate the distance between the clusters and the BWP

value for measuring the fitness of a chromosome. Therefore, the $b(i, j)$ of the new algorithm design proposed in this paper keeps the original calculation method unchanged, and $w(i, j)$ is simplified to calculate the shortest distance from the center of the original cluster to the center of other clusters. If the clustering effect is good enough and the distance within the group is tight, the distance between the clusters to the other clusters is approximately equal to the distance between the discrete points closest to the other clusters to the other clusters.

Table 1. Info of data sets

Data sets	Number of attributes	Number of categories	Number of instances
Iris	4	3	150
Wine	13	3	178
Soybean	35	4	47
Seeds	7	3	210

Table 2. Parameter table for clustering experiments of Iris

Parameter name	Value	Parameter name	Value
Running times	10	Function set	||, &&
Number of generations	100	Terminal set	x1–x150
Size of groups	100	Mutation probability	0.4
Head length	2	Single point recombination probability	0.7
Tail length	3	Double point recombination probability	0.7

4 Experiment and Result Analysis

The parallel GEP clustering algorithm based on producer-consumer model (PGE PC/PCM) proposed in this paper is implemented by Java language. This paper compares the parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) and the basic gene expression programming clustering algorithm. And it also uses these two algorithms to compare the average accuracy, the highest accuracy, the running time, the average correct clustering document, the highest correct clustering document and other evaluation indicators of the four data sets.

4.1 Data Sets

The experiment in this paper uses 4 data sets. The data sets are derived from the UCI database for machine learning (http://archive.ics.uci.edu/ml/) proposed by the University of California Irvine. The four data sets are Iris, Wine, Soybean, and Seeds. The main info of four data sets is listed in Table 1.

4.2 Parallel GEP Clustering Algorithm Based on Producer-Consumer Model (PGEPC/PCM)

In order to analyze the influence of the parallelization method based on the producer-consumer model on the GEP clustering algorithm, this paper carried out the parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) at first, and cluster for the above 4 data sets. The algorithm running environment is windows operating system and Java 1.7, using Redis as a tool. The experimental parameter settings are shown in Tables 2, 3, 4 and 5.

Table 3. Parameter table for clustering experiments of Wine

Parameter name	Value	Parameter name	Value
Running times	10	Function set	||, &&
Number of generations	100	Terminal set	x1–x178
Size of groups	100	Mutation probability	0.4
Head length	2	Single point recombination probability	0.7
Tail length	3	Double point recombination probability	0.7

Table 4. Parameter table for clustering experiments of Soybean

Parameter name	Value	Parameter name	Value
Running times	10	Function set	||, &&
Number of generations	100	Terminal set	x1–x47
Size of groups	100	Mutation probability	0.4
Head length	3	Single point recombination probability	0.7
Tail length	4	Double point recombination probability	0.7

Table 5. Parameter table for clustering experiments of Seeds

Parameter name	Value	Parameter name	Value
Running times	10	Function set	||, &&
Number of generations	100	Terminal set	x1–x210
Size of groups	100	Mutation probability	0.4
Head length	2	Single point recombination probability	0.7
Tail length	3	Double point recombination probability	0.7

Table 6. Clustering experiment results of Iris

Data set	Running times	Highest accuracy		Run time/s	
		GEP	PGEPC/PCM	GEP	PGEPC/PCM
Iris	1	89.33	89.33	445	234
	2	88	89.33	435	245
	3	88	90.66	479	250
	4	90.66	90.66	470	247
	5	88	89.33	430	240
	6	90.66	90.66	447	255

The four data sets were clustered based on the above parameter settings, and the experimental results are shown in Tables 6, 7, 8 and 9.

The comparison of the running time and the highest accuracy of the parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) clustering and the basic GEP clustering is shown in Figs. 2, 3, 4 and 5.

It can be compared from Fig. 2, 3, 4 and 5, the parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) has no visible accuracy improvement in each data set, but it has improved a lot in the running

Table 7. Clustering experiment results of Wine

Data set	Running times	Highest accuracy		Run time/s	
		GEP	PGEPC/PCM	GEP	PGEPC/PCM
Wine	1	87.07	87.64	683	308
	2	87.64	87.07	678	309
	3	83.70	87.07	664	310
	4	87.07	87.07	679	305
	5	87.07	87.64	684	312
	6	87.64	87.07	664	315

Table 8. Clustering experiment results of Soybean

Data set	Running times	Highest accuracy		Run time/s	
		GEP	PGEPC/PCM	GEP	PGEPC/PCM
Soybean	1	85.10	85.10	85.10	85.10
	2	85.10	87.23	87.23	87.23
	3	85.10	87.23	87.23	87.23
	4	87.23	87.23	87.23	87.23
	5	87.23	87.10	87.10	87.10
	6	87.10	87.10	87.10	87.10

Table 9. Clustering experiment results of Seeds

Data set	Running times	Highest accuracy		Run time/s	
		GEP	PGEPC/PCM	GEP	PGEPC/PCM
Seeds	1	78.57	78.57	803	515
	2	79.04	78.57	789	504
	3	78.57	78.57	794	512
	4	78.57	78.57	782	518
	5	79.04	79.04	795	519
	6	78.57	78.57	785	514

speed. It can be concluded that this parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) improves the running speed of the algorithm without affecting the change of the overall fitness, thus improving the performance of the algorithm, and the improvement is visible.

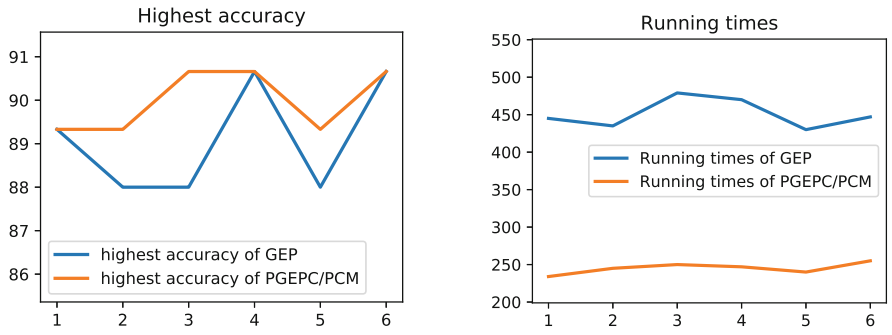

Fig. 2. Clustering experiment result comparison figure of Iris

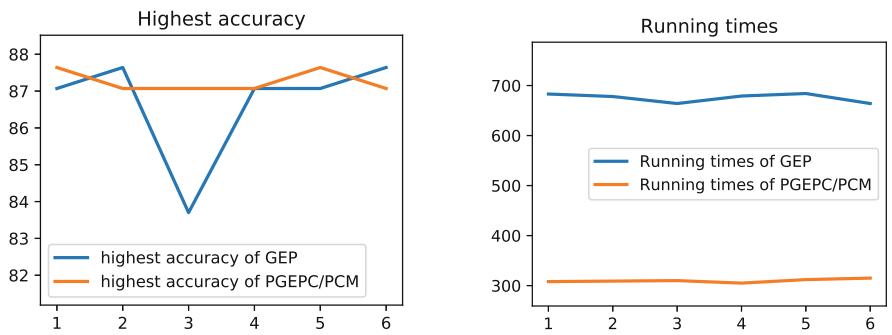

Fig. 3. Clustering experiment result comparison figure of Wine

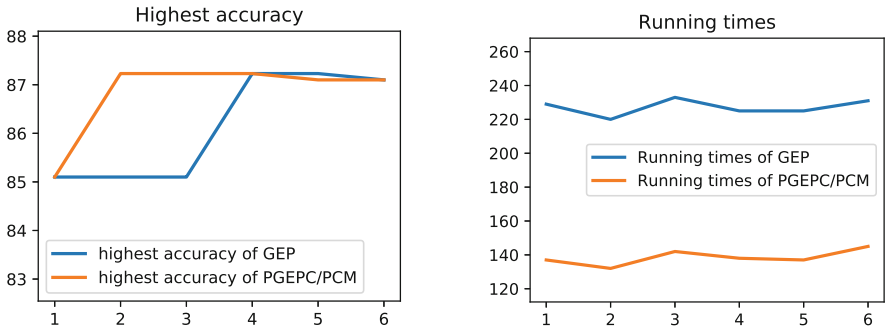

Fig. 4. Clustering experiment result comparison figure of Soybean

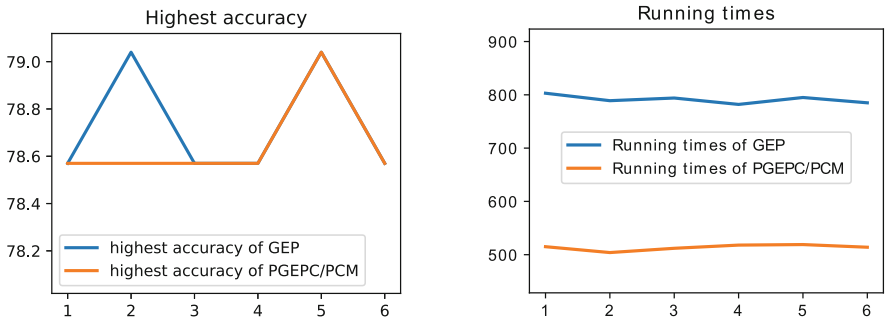

Fig. 5. Clustering experiment result comparison figure of Seeds

5 Conclusion and Future Work

In order to get better clustering results, in this paper, we have conducted analysis and research to optimize the evolutionary operators and algorithm flows based on GEP for cluster analysis. Aiming at the shortcomings of the basic GEP clustering algorithm, a parallel GEP clustering algorithm based on producer-consumer model (PGEPC/PCM) proposed, which parallelizes the time-consuming operations such as fitness calculation, recombination and mutation in GEP clustering analysis to speed up, improves the calculation method of fitness function to enable it to cluster automatically. Extensive experiments on 4 widely used benchmark Iris, Wine, Soybean and Seeds from the UCI machine learning data sets are conducted to investigate the influence of algorithmic components and results are compared with the traditional GEP clustering algorithm. These comparisons demonstrate the competitive efficiency of the proposed algorithm.

Then there are several areas worth studying. For example, The method of fitness calculation has not been perfected and the problem of local optimization has not been solved, which will be the follow-up work of this paper, how to set the head coding of chromosome reasonably and the selection of weights needs further study.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 61573157 and 61703170), Science and Technology Project of Guangdong Province of China (Grant Nos. 2018A0124 and 2017A020224004), Science and Technology Project of Tianhe District of Guangzhou City (Grant No. 201702YG061), Science and technology innovation project for College Students (Grant No. 201910564129). The authors also gratefully acknowledge the reviewers for their helpful comments and suggestions that helped to improve the presentation.

References

1. Ferreira, C.: Gene Expression Programming. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-32849-1
2. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on k-means algorithm for optimal clustering in RN. Inf. Sci. **146**(1–4), 221–237 (2002). https://doi.org/10.1016/s0020-0255(02)00208-6
3. Chen, Y., Tang, C., Ye, S.Y., Li, C., Liu, Q.H.: An auto-clustering algorithm based on gene expression programming **39**, 107–112 (2007)
4. Colbourn, E., Roskilly, S., Rowe, R., York, P.: Modelling formulations using gene expression programming - a comparative analysis with artificial neural networks. Eur. J. Pharm. Sci. **44**(3), 366–374 (2011). https://doi.org/10.1016/j.ejps.2011.08.021
5. Deng, S., Zhou, A.H., Yue, D., Hu, B., Zhu, L.P.: Distributed intrusion detection based on hybrid gene expression programming and cloud computing in a cyber physical power system. IET Control Theory Appl. **11**(11), 1822–1829 (2017). https://doi.org/10.1049/iet-cta.2016.1401
6. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. ArXiv cs.AI/0102027 (2001)
7. Gholami, A., Bonakdari, H., Zeynoddin, M., Ebtehaj, I., Gharabaghi, B., Khodashenas, S.R.: Reliable method of determining stable threshold channel shape using experimental and gene expression programming techniques. Neural Comput. Appl. **31**(10), 5799–5817 (2018). https://doi.org/10.1007/s00521-018-3411-7
8. Hongguo, C., Chanan, Y.: Research on GEP-based cluster algorithm for serial program to be parallelized. J. S.-Cent. Univ. Nationalities (Nat. Sci.Ed.) **4**, 112–115 (2017)
9. Huang, Z., Li, M., Chousidis, C., Mousavi, A., Jiang, C.: Schema theory-based data engineering in gene expression programming for big data analytics. IEEE Trans. Evol. Comput. **22**(5), 792–804 (2018). https://doi.org/10.1109/tevc.2017.2771445
10. Jedrzejowicz, J., Jedrzejowicz, P., Wierzbowska, I.: Implementing gene expression programming in the parallel environment for big datasets' classification. Vietnam J. Comput. Sci. **06**(02), 163–175 (2019). https://doi.org/10.1142/s2196888819500118
11. Jiang, D.H., Zhang, S.Y.: K-means auto-clustering algorithm based on gene expression programming. Comput. Simul. **27**, 216–220 (2010)
12. Jiang, Z., Li, T., Min, W., Qi, Z., Rao, Y.: Fuzzy c-means clustering based on weights and gene expression programming. Pattern Recogn. Lett. **90**, 1–7 (2017). https://doi.org/10.1016/j.patrec.2017.02.015
13. Murthy, C., Chowdhury, N.: In search of optimal clusters using genetic algorithms. Pattern Recogn. Lett. **17**(8), 825–832 (1996). https://doi.org/10.1016/0167-8655(96)00043-8

14. Yang, L., Li, K., Zhang, W., Zheng, L., Ke, Z., Qi, Y.: Optimization of classification algorithm based on gene expression programming. J. Ambient Intell. Humanized Comput. (2017). https://doi.org/10.1007/s12652-017-0563-8
15. Zheng, Y., Jia, L., Cao, H.: Multi-objective gene expression programming for clustering. Inf. Technol. Control **41**(3) (2012). https://doi.org/10.5755/j01.itc.41.3.1330
16. Zhong, J., Feng, L., Ong, Y.S.: Gene expression programming: a survey [review article]. IEEE Comput. Intell. Mag. **12**(3), 54–72 (2017). https://doi.org/10.1109/mci.2017.2708618
17. Zhong, J., Ong, Y.S., Cai, W.: Self-learning gene expression programming. IEEE Trans. Evol. Comput. **20**(1), 65–80 (2016). https://doi.org/10.1109/tevc.2015.2424410
18. Aytac, G., Ali, A.: New approach for stage-discharge relationship: gene-expression programming. J. Hydrol. Eng. **14**(8), 812–820 (2009). https://doi.org/10.1061/(ASCE)HE.1943-5584.0000044