Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 10, October 2019, pp. 2667-2674
DOI: 10.23940/ijpe.19.10.p12.26672674

Automatic Software Testing Target Path Selection using K-Means Clustering Algorithm

Yan Zhang[a], Li Qiao[b], Xingya Wang[b,*], Jingying Cai[a], and Xuefei Liu[c]

[a]*Department of Computer and Information Technology, Mudanjiang Normal University, Mudanjiang, 157012, China*
[b]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210093, China*
[c]*Department of Computer Science and Technology, Heilongjiang University, Harbin, 150080, China*

Abstract

Path testing is an effective method of software testing. It is not realistic to achieve coverage for all paths during complex software testing. Selecting the correct paths as target paths is a key problem. A method of selecting target paths based on the K-means algorithm is presented in this study. First, we divide paths into different groups using the K-means algorithm, so that paths having high similarity are divided into the same group. Then, we choose the cluster centers as targets and ensure that the selected target paths have more considerable differentiation, which guarantees the adequacy of later testing. The experimental results demonstrate the effectiveness of the proposed method.

Keywords: path coverage; software testing; K-means; clustering center; test adequacy

(Submitted on September 15, 2019; Revised on October 16, 2019; Accepted on October 20, 2019)

1. Introduction

Software testing is an essential means to ensure software quality [1]. It is the process of assessing whether a software can meet the expected requirements and verify the quality of software [2]. Path coverage is a white box test method with high coverage and strong error detection capability. Shan et al. proposed that many problems of software testing involve generating test data for path coverage. These problems can be described as follows: for a target path of a program under testing, search for a test datum in the input domain of the program so that the traversed path of the test datum is only the target one [3]. In the case of a large number of complex software paths to be tested, it is unrealistic to achieve full path coverage. At this point, the paths that are selected as the target ones directly affect the quality of the test data. At present, some scholars have studied the selection method of target paths. For example, Jiang et al. proposed a C# change influence path set generation method based on function call path, but this method is mainly used for the path selection of regression tests [4]. Path coverage testing mostly manually determines the target path, a method that is related to the tester's experience and takes time to distinguish the similarity between paths. Moreover, the quality of the test data corresponding to these paths is difficult to guarantee. If the appropriate method is adopted, selecting part of paths with a certain degree of discrimination as the target ones is undoubtedly helpful for generating high-quality test data; if the selection process is automatically implemented, the test efficiency will be effectively improved.

In path coverage testing, if the selected target paths can be ensured to have more significant discrimination between each other, it is more conducive to achieving higher test coverage of the tested program, thereby achieving higher test sufficiency [5]. In this paper, the K-means clustering method is used to group many paths, the paths within each group have certain similarities, and the paths between different groups have greater discrimination. In each group, the path closest to the cluster center is selected as the target path. We only need to use the selected target paths to generate test data, which greatly reduces the test workload. Moreover, the target paths selected after clustering are representative of each other to ensure the quality of the test.

* Corresponding author.
E-mail address: xingyawang@nju.edu.cn

2. K-Means Clustering Method

Means clustering is an unsupervised hard partitioning clustering method [6]. The objective is to find *k* clusters from the data based on the objective function F given in Equation (1).

$$F = \sum_{i=1}^{K} \sum_{j=1}^{N_i} d^2(C_i, X_j) \tag{1}$$

Where $d^2(C_i, X_j)$ is the squared Euclidean distance between the i^{th} cluster centroid and the j^{th} data point. N is the total number of data points, N_i is the amount of data in the i^{th} cluster, and $\sum_{i=1}^{k} N_i = N$. Based on the distance obtained, the points are assigned to the cluster with minimum distance from the centroid.

After the points are clustered, the mean of all points belonging to the cluster is found. Then mean value is assigned as the new cluster centroid for the next iteration. This process is repeated until the centroid obtained is the same as that of the previous iteration. The aim of the K-means algorithm is to minimize the objective function.

The K-means algorithm above takes the Euclidean distance as the proximity measure. The sum of squares of errors is used as the objective function to measure the quality of clustering. In fact, the K-means algorithm is not limited to the Euclidean distance. Tan et al. gave the measurement method of the document data and cosine similarity measure and proposed that the K-means algorithm can be used for many types of data [7].

3. Path Clustering based on K-Means

The K-means algorithm is used to cluster the path of the program under testing. The selection of the cluster center and the distance calculation method between the paths and the center path need to be very different from other data clusters. For the practical problem of path coverage testing, it is necessary to design a method for path representation that facilitates comparison between different paths, establish an objective function that distinguishes the distance between different paths, and give a method for determining the cluster center in the iterative process of the algorithm.

This section provides an overview of the proposed approach to determine the path cluster of the program under testing based on K-means. It details three important steps: path representation, path discrimination, and determination of cluster centers.

3.1. Method for Path Representation

In order to reduce the amount of computation during clustering, we use our previous research results to represent the program under testing as a binary tree and then encode the path of the program under testing based on Huffman coding [8]. For the selection branch in the program, the loop statement only considers the case of executing once and not executing and encodes the path according to the Huffman coding method. The feature is that the path coding is composed of 0 and 1 codes, and the path coding sequence can reflect the branch direction of the path. It is prefix coding, that is, each path coding is not a prefix of other codes. Since this kind of coding only considers branch statements, the path length is less than other coding modes such as statement number, and the calculation amount is small when clustering.

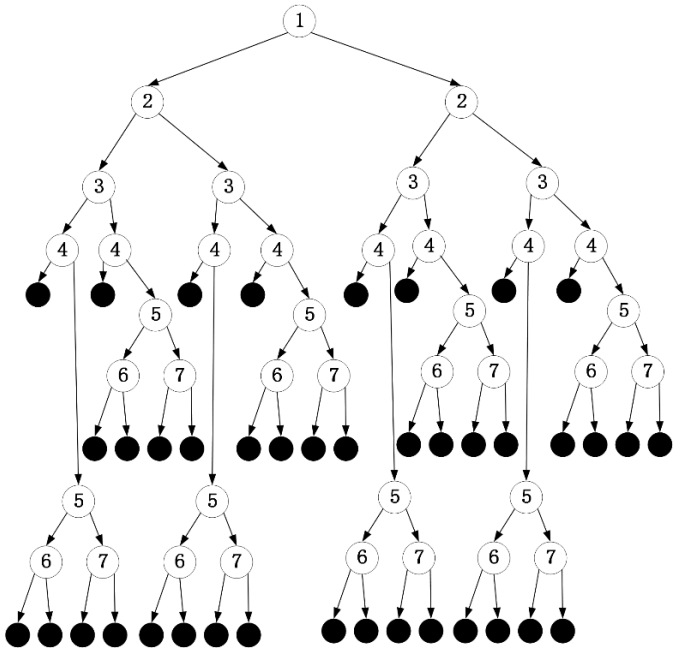
As shown in Figure 1(a), the triangle classification procedure from reference [9] is used, and the branch statements are labeled 1 to 7. Figure 1(b) is a binary tree corresponding to the source program in Figure 1(a), and all the leaf nodes in the tree represent one path. Using the Huffman coding method, the left branch of the binary tree is coded as "1", and the right branch is coded as "0". In Figure 1(b), the path corresponding to the left branch of the leftmost node 4 is coded as "1111", and the path corresponding to the left branch node of the leftmost node 6 is coded as "111011". The path corresponding to the right branch of the rightmost node 7 is encoded as "000000". Thus, the length of the path is greatly shortened, and the workload of calculating discrimination is effectively reduced.

3.2. Path Discrimination

The simple matching coefficient [7] is used to calculate the similarity between paths, as follows:

```
void trangle(int a,int b,int c)
  {  int type,t;
  1  if(a>=b)
        {  t=a;a=b;b=t;  }
  2  if(a>=c)
        {  t=a;a=c;c=t;  }
  3  if(b>=c)
        {  t=b;b=c;c=t;  }
  4  if(a+b<=c)
        type=4;
     else
  5     {  if(a==b)
  6        {  if(b==c)
                type=1;
             else
                type=2; }
          else
  7       {  if(b==c)
                type=2;
             else
                type=3; }
        }
     return  type;
  }
```

(a) Source program



(b) Binary tree

Figure 1. Triangle classification program and its binary tree

Let x_p and x_q denote two paths of the program under testing. By comparing the codes of the two paths from front to back, the following four quantities can be obtained:

f_{00} represents the number of coded bits where x_p takes 0 and x_q takes 0;
f_{01} represents the number of coded bits where x_p takes 0 and x_q takes 1;
f_{10} represents the number of coded bits where x_p takes 1 and x_q takes 0;
f_{11} represents the number of coded bits where x_p takes 1 and x_q takes 1.

The path similarity is expressed as

$$SMC(x_p, x_q) = \frac{\text{The number of same codes}}{\text{The length of the shortest path}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}} \tag{2}$$

Then, $1 - SMC(x_p, x_q)$ indicates the discrimination between the paths x_p and x_q, denoted by $f(x_p, x_q)$, as follows:

$$f(x_p, x_q) = 1 - SMC(x_p, x_q) = 1 - \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}} = \frac{f_{01} + f_{10}}{f_{01} + f_{10} + f_{11} + f_{00}} \tag{3}$$

For example, for the program under testing in Figure 1, the discrimination between the paths "1111" and "111011" is

$$f(\text{"1111"},\text{"111011"}) = \frac{f_{01} + f_{10}}{f_{01} + f_{10} + f_{11} + f_{00}} = \frac{0+1}{0+1+3+0} = 0.25$$

The discrimination between the paths "1111" and "000000" is

$$f(\text{"1111"},\text{"000000"}) = \frac{f_{01} + f_{10}}{f_{01} + f_{10} + f_{11} + f_{00}} = \frac{0+4}{0+4+0+0} = 1$$

3.3. Determination of Cluster Centers

When the K-means algorithm clusters numerical data, the average value is selected as the class center of the newly generated class. However, path clustering does not cluster numerical data and requires special processing. The method designed in this paper is as follows: for each class, calculate the discrimination between each path and other paths, form a discrimination matrix, and sum each column in the matrix. The path whose sum of discrimination between it and the other paths is the smallest is selected as the new cluster center.

Assume that there are m paths in the i^{th} class, the discrimination matrix (denoted as DM) is a matrix of $m \times m$, and then

$$DM(i) = \begin{bmatrix} f(x_1, x_1) & f(x_1, x_2) & \cdots & f(x_1, x_m) \\ f(x_2, x_1) & f(x_2, x_2) & \cdots & f(x_2, x_m) \\ \vdots & \vdots & \ddots & \vdots \\ f(x_m, x_1) & f(x_m, x_2) & \cdots & f(x_m, x_m) \end{bmatrix}$$

Each element $f(x_p, x_q)$ in matrix DM is calculated according to Equation (2), where $1 \le p \le m$, $1 \le q \le m$ and $p \ne q$. When $p = q$, $f(x_p, x_q) = 0$, which means the main diagonal elements of the discrimination matrix are all zero.

It can be determined from Equation (2) that $f(x_p, x_q) = f(x_q, x_p)$. The above matrix is a symmetric matrix, and the values in the discrimination matrices between m paths can be calculated $m(m-1)/2$ times.

Calculate the sum of each column element, and the sum of the q^{th} column is denoted as $sum(q)$. Then,

$$sum(q) = \sum_{p=1}^{m} f(x_p, x_q)$$

Select the path corresponding to the column with the smallest value of the sum of the elements in the m column as the new class center.

Still taking the program in Figure 1 as an example, suppose there are five paths in the second class, which are $x_1 = \text{"1111"}$, $x_2 = \text{"111011"}$, $x_3 = \text{"1011"}$, $x_4 = \text{"101010"}$, and $x_5 = \text{"110011"}$. Then, the discrimination matrix is

$$DM(2) = \begin{bmatrix} 0.00 & 0.25 & 0.25 & 0.50 & 0.50 \\ 0.25 & 0.00 & 0.50 & 0.33 & 0.17 \\ 0.25 & 0.50 & 0.00 & 0.25 & 0.75 \\ 0.50 & 0.33 & 0.25 & 0.00 & 0.50 \\ 0.50 & 0.17 & 0.75 & 0.50 & 0.00 \end{bmatrix}$$

Sum each column separately to get $sum(1) = 1.5$, $sum(2) = 1.25$, $sum(3) = 1.75$, $sum(4) = 1.58$, and $sum(5) = 1.92$. The minimum value is $sum(2) = 1.25$. Thus, choose the second path $x_2 = \text{"111011"}$ as the new class center.

3.4. Algorithm Description

Input: N paths of the program under test, the number of target paths to be selected, k; Output: k target paths.

Steps:

(1) Randomly select k paths from the N paths as the initial class center;

(2) Calculate the degree of discrimination between the other $N-K$ paths and these central objects according to Equation (3). Divide each path with the class center that has the smallest degree of discrimination, and complete the division of all paths.

(3) Recalculate the central path of each cluster (with change);

(4) Repeat steps (2)-(3) until each class center remains unchanged.

4. Experiment

4.1. Experimental Setup

In order to verify the validity of the proposed method, the program in reference [9], which is shown in Figure 1, is selected as the program to be tested, and the path is encoded according to the method of [8]. There are 40 paths, and the corresponding codes are shown in Table 1. Set the number of target paths to four, five, and eight, and count the times of experiment iteration, the required time, and the number of target paths. Since the target path is not selected to be more mature, the random method is used to compare with the method. In order to avoid the error caused by the random selection of the initial cluster center, the experiment is repeated 15 times, and the average result is taken. The sum of the degrees of discrimination between the coding structures of the selected target paths and the coverage of the representative target paths corresponding to the triangle types are compared.

Table 1. Paths of the program under testing

No. of paths	Path coding	No. of paths	Path coding	No. of paths	Path coding	No. of paths	Path coding
1	1111	11	1011	21	0111	31	0011
2	111011	12	101011	22	011011	31	001011
3	111010	13	101010	23	011010	33	001010
4	111001	14	101001	24	011001	34	001001
5	111000	15	101000	25	011000	35	001000
6	1101	16	1001	26	0101	36	0001
7	110011	17	100011	27	010011	37	000011
8	110010	18	100010	28	010010	38	000010
9	110001	19	100001	29	010001	39	000001
10	110000	20	100000	30	010000	40	000000

4.2. Experimental Results and Analysis

Table 2 shows the results of 15 experiments when the number of target paths is set to five. The "No. of paths" column gives the number of selected target paths after clustering in Table 1, and the triangle type corresponding to the path can be seen from the corresponding path coding. For example, for the second experimental result corresponding to the second row in the table, the path of No. 25 is "011000" in Table 1, and the first three digits of the path encoding "011" indicate the branch direction corresponding to the three-number order. The code after the fourth bit represents the type of triangle. The path of No. 25 is followed by the "000" corresponding to a non-equilateral triangle. The "010" following the path "101010" of No. 13 corresponds to an isosceles triangle. The "1" following the path "1011" of No. 11 cannot constitute a triangle. The "001" following the path "110001" of No. 9 corresponds to an isosceles triangle. In this way, the five target paths selected during the second time all contain four kinds of results corresponding to three numbers. The "Total types" column in the table represents the number of triangle types corresponding to the selected target path. The larger the value, the more comprehensive the path covered by the selected path. The "Total Discrimination" column calculates the sum of discrimination according to Equation (2) between each of the selected target paths. The larger the value, the greater the difference between the selected target paths.

As can be observed from Table 2, when clustering paths using the proposed method, the average value of the numbers of selected target paths for the 15 experiments is 3.533, which is larger than that of the random method (3.200). The average of total discrimination is 11.222, which is also larger than that of the random method (10.422). It is proven that the target paths selected by the K-means based clustering method have good discrimination between each other. Because the paths traversed by different data have a great degree of discrimination, when the large degree of discrimination is reflected in the

path type, there is also better coverage. Of the 15 experiments using the method of this paper, nine reach the four types of paths corresponding to the triangle classification procedure. However, in the 15 experiments of the random method, only five selected paths include all four types. This shows that the target path selected by this method can achieve better structural coverage. The test data corresponding to these paths will be more conducive to the discovery of defects in subsequent tests. The standard deviation of the number of path types and the total discrimination degree of the method in this paper are less than those of the random method, which shows that our method achieves a certain stability.

Table 2. Experimental results when the number of target paths *K* is five

No.	Our method			Random method		
	No. of paths	Total types	Total discrimination	No. of paths	Total types	Total discrimination
1	20, 3, 40, 8, 34	2	10.000	20, 3, 40, 8, 34	2	9.333
2	25, 37, 13, 11, 9	4	11.667	25, 37, 13, 11, 9	4	11.667
3	31, 19, 24, 3, 27	3	11.500	31, 19, 24, 23, 27	3	10.667
4	8, 32, 14, 2, 26	3	11.333	8, 32, 14, 2, 26	3	11.333
5	19, 6, 7, 5, 35	4	10.500	20, 6, 8, 5, 35	3	9.167
6	11, 7, 26, 37, 23	3	11.667	16, 7, 26, 38, 23	3	10.667
7	3, 27, 20, 11, 25	4	11.500	7, 27, 20, 15, 25	2	9.333
8	13, 17, 24, 20, 26	4	12.000	13, 17, 19, 20, 26	4	11.167
9	5, 23, 12, 6, 4	4	9.667	5, 22, 12, 6, 4	4	9.333
10	8, 39, 31, 37, 22	3	10.333	8, 39, 31, 37, 22	3	10.333
11	40, 20, 32, 6, 24	4	11.667	40, 20, 32, 6, 24	4	11.667
12	26, 23, 32, 36, 15	4	11.667	26, 23, 33, 36, 15	3	11.000
13	27, 40, 4, 16, 14	4	11.000	29, 40, 4, 16, 14	3	10.000
14	31, 1, 22, 8, 10	4	11.667	25, 1, 22, 8, 10	4	9.000
15	30, 11, 7, 37, 25	3	12.167	30, 11, 27, 37, 25	3	11.667
Average value	-----	**3.533**	**11.222**	-----	**3.200**	**10.422**
Standard deviation	-----	0.618	0.727	-----	0.653	0.960

The number of target paths is set to four and eight respectively, and the experimental results are compared with the experimental results of the five target paths. The results are shown in Table 3.

Table 3. Comparison of experimental results for different target path numbers

K	Our method				Random method			Our method/random method	
	Mean number of iterations	Average value of types	Total discrimination		Average value of types	Total discrimination		Average value of types	Average value of total discrimination
			Average value	Standard deviation		Average value	Standard deviation		
4	1.667	3.067	7.033	0.698	2.733	6.355	0.807	**1.122**	**1.107**
5	1.733	3.533	11.222	0.727	3.200	10.422	0.960	**1.104**	**1.077**
8	1.800	3.733	28.855	0.926	3.467	27.233	1.558	**1.076**	**1.059**

From Table 3, we can see that among the experimental results of the three kinds of target paths, the method of this paper is superior to the random method in terms of discrimination and type coverage. Both of these aspects can be seen in the ratio of the method of this paper to the random method: as the number of target paths ranges from 4 to 5 and then to 8, the ratio of coverage types changes from 1.122 to 1.104 and then to 1.076, showing a downward trend. The total discrimination also drops from 1.107 to 1.077 and then decreases to 1.059, which indicates that the advantage of this method is more obvious when the number of target paths is smaller. As the number of target paths increases, the number of cluster iterations required to complete path selection increases. However, most experiments can complete clustering in the second generation, indicating that the algorithm is relatively easy to implement.

Moreover, from the standard deviation, the standard deviation of the method in the total number of path types and the discrimination degree is smaller than that of the random method, which indicates that the method has good stability.

The total discrimination is better able to reflect the difference between the selected target paths. In order to more scientifically verify that the target path selected by this method has better discrimination, we use the important method of statistical inference - hypothesis test [10] to analyze the experimental results.

Let X_1, X_2 denote the total discrimination of each path selected by the method and the random method, respectively. For the convenience of description, the same method uses the same symbol for the comparison of the path discrimination of the experimental results of different path numbers, and then X_1, X_2 are random variables. In addition, because the values of

X_1, X_2 are affected by many random factors, they obey a normal distribution in many experiments. Let $X_i \sim N(\mu_i, \sigma_i^2)$, $i = 1, 2$. Compare the mean of the random variables corresponding to different methods, that is, the size of $\mu_i (i = 1, 2)$. The larger the value of μ_i, the higher the number of target path types selected by the method, and the better the method.

In the following, taking the target path number $K = 5$ as an example, the detailed process of comparing μ_1 and μ_2 with a hypothesis test is given. Consider that the sample variance is an unbiased estimate of the population variance. Therefore, the value of the sample variance is used as an estimate of the population variance, that is, the value of the sample standard deviation is used as an estimate of the population standard deviation to obtain $\sigma_1 = 0.727$ and $\sigma_2 = 0.960$. Here, the sample size $n_1 = n_2 = 15$, $\bar{X}_1 = 11.222$, $\bar{X}_2 = 10.422$, the value of the significance level α is 0.01, and $Z_\alpha = 2.325$ for the two-sided test.

Step 1 Establish the original hypothesis $H_0 : \mu_1 \le \mu_2$ and the opposing hypothesis $H_1 : \mu_1 > \mu_2$;

Step 2 Construct the statistic $U_1 = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}}}$;

Step 3 Give the rejection domain $U_1 = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}}} \ge Z_\alpha$;

Step 4 Calculate the value of the statistic ;

Step 5 Give the conclusion: because $U_1 \approx 2.572 > Z_\alpha = 2.325$ falls within the rejection region, reject H_0 and accept H_1. In other words, the expected value of the total discriminant of the path selected by this method is obviously larger than that of the random method. This shows that the target path selected by this method is more representative than the random method.

At the same significance level $\alpha = 0.01$, for the other two path numbers $K = 4$ and $K = 8$, the total discrimination in the experimental results is statistically analyzed. The results are listed in Table 4.

Table 4. Hypothesis test results

Number of target paths	Test items	Value of the statistic	Conclusion
$K = 4$	Total discrimination	$U_1 = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}}} = \dfrac{7.033 - 6.355}{\sqrt{\dfrac{0.689^2}{15} + \dfrac{0.807^2}{15}}} \approx 2.462 > 2.325$	Reject H_0, accept H_1, i.e., $\mu_1 > \mu_2$
$K = 8$	Total discrimination	$U_1 = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}}} = \dfrac{28.855 - 27.233}{\sqrt{\dfrac{0.926^2}{15} + \dfrac{1.558^2}{15}}} \approx 3.466 > 2.325$	Reject H_0, accept H_1, i.e., $\mu_1 > \mu_2$

It can be seen from Table 4 that for the other two kinds of path selection results, the total path discrimination of this method is higher than that of the random method.

5. Conclusions

This paper is aimed at the selection of target paths and attempts to cluster target paths using the K-means clustering method. In the process of clustering, a calculation method of path discrimination based on program structure and a cluster center selection method are designed, which effectively realizes the clustering of paths and the selection of target paths. This makes the selected target paths have a good degree of discrimination. From the reflection to the

program structure, the selected target path has good coverage which is beneficial to improve the effectiveness of the path test.

It should be noted that the evaluation of method performance in this paper needs to be verified in more complicated industrial procedures. In addition, the representation of the binary tree of the program under testing and how to automate the path generation are also needed for further research.

Acknowledgements

This study was jointly funded by the National Natural Science Foundation of China (No. 61573362), Heilongjiang Natural Science Foundation (No. F2016039), Heilongjiang Provincial Department of Education Basic Research Business Expenses Project (No. 2018-KYYWF-0419, 1353MSYYB005), Jiangsu Planned Projects for Postdoctoral Research Funds (No. 2018K028C), Innovation Project for State Key Laboratory for Novel Software Technology (No. ZZKT2018B02), Mudanjiang Science and Technology Plan Project (No. Z2016s0027), and Mudanjiang Normal University Research Project (No. GP201602, FD2014001, QY2014001, KB0263)

References

1. Y. Zhang and D. W. Gong, "Evolutionary Generation of Test Data for Paths Coverage based on Scarce Data Capturing," *Chinese Journal of Computers*, Vol. 36, No. 8, pp. 2429-2440, December 2013
2. S. Y. Wang, J. J. Juan, and J. Z. Sun, "Reduction Method of Test Suites based on Weak Mutation Criteria," *Journal of Computer Applications*, Vol. 39, No. 2, pp. 534-539, September 2018
3. J. Shan, Y. Jiang, and P. Sun, "Research Progress in Software Testing," *Acta Scientiarum Naturalum Universitis Pekinensis*, Vol. 41, No. 1, pp. 134-145, January 2005
4. J. C. Jiang and Y. M. Mu, "A Generation Method of Path Set with Change Impact for C# based on FCP," *Journal of Beijing Information Science and Technology University*, Vol. 33, No. 3, pp. 21-25, June 2018
5. H. S. Li and R. Guo, "Software Testing Technology Case Course," Tsinghua University Press, Beijing, April 2012
6. S. Majhi and S. Biswal, "Optimal Cluster Analysis using Hybrid K-Means and Ant Lion Optimizer," *Karbala International Journal of Modern Science*, Vol. 4, No. 4, pp. 347-360, 2018
7. P. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," Posts and Telecom Press, 2017
8. D. W. Gong and Y. Zhang, "Novel Evolutionary Generation Approach to Test Data for Multiple Paths Coverage," *Acta Electronica Sinica*, Vol. 38, No. 6, pp. 1299-1304, June 2010
9. Y. Cao, C. Hu, and L. Li, "An Approach to Generate Software Test Data for a Specific Path Automatically with Genetic Algorithm," in *Proceedings of the 8th International Conference on Reliability, Maintainability and Safety*, pp. 888-892, Cheng Du, China, September 2009
10. P. C. Gou, Y. Zhao, H. G. Yi, J. L. Bai, H. Yu, and F. Chen, "The Application of Permutation Test in the Hypothesis Test," *Application of Statistics and Management*, Vol. 25, No. 5, pp. 616-622, September 2006