Eng Int Syst (2025) 6: 611–619

Engineering
Intelligent Systems

Computer Software Fault Location Method Based on Machine Vision and Image Processing

Wei Yan[1,a,*] **and Chen Yang**[2,b]

[1] *Police Information Department, Liaoning Police College, Dalian 116036, Liaoning, China*
[2] *School of Software, Dalian University of Foreign Languages, Dalian 116044, Liaoning, China*

With the rapid development of science and technology, software applications have penetrated various industries, and the requirements for software systems have become increasingly higher. A variety of software systems have gradually changed people's lives and their economies. When people demand more and more software, software failures tend to occur. In order to reduce the harm caused by software errors, this paper proposes a computer software fault location method based on machine vision and image processing. First, an analysis was conducted of the graph mining technology, and then an experiment was conducted to compare the two different graph reduction methods in series and parallel to determine their fault location efficiency. According to the experimental results, when the number of nodes was small, the time consumed by serial reduction in fault location was shorter. However, when the number of nodes increased, the advantages of parallel reduction became increasingly obvious. When the number of nodes was 5646, the efficiency reached 55.22%, which significantly improved the efficiency of fault location. This can help programmers to locate and repair the application faults, improve the performance of software security, and reduce the loss caused by software failure.

Keywords: software fault location; machine vision; image processing; graph mining technology

1. INTRODUCTION

With the rapid development of information technology, Internet applications have penetrated various industries, not only changing people's lives and living standards, but also boosting the development of the entire economy. At the same time, people's daily lives have increasingly become strongly dependent on these applications. In many ways, these applications are closely related to several aspects of people's daily lives, including safety. Therefore, the reliability of software in many respects can have a direct impact on some areas of society. In practice, software failures are commonplace, which has caused adverse effects on and great difficulties for the production and quality of software. Because of the huge danger that can be posed by software faults, the demand for software fault location has emerged. It is crucial to reduce the cost of software testing, ensure the reliability of software, quickly and accurately find the fault in the system, and establish a set of highly reliable fault location systems, thus improving the accuracy and effectiveness of fault diagnosis.

When a software failure occurs, it can have a huge impact on the overall quality of the software, make the entire system invalid, and cause huge economic losses. Naidu and Ashok [1] proposed a two-terminal transmission line fault location method based on traveling wave. This method used the asynchronous current measurements of intelligent electronic equipment recorded at both ends. In view of the difficulty of locating high-resistance grounding fault, Lan et al. proposed a fault location method for two-terminal asynchronous High Voltage Direct Current (HVDC) transmission line based on Hilbert-Huang transform and

*Corresponding Author. [a]Email: yanweiln@126.com, [b]yangchenyjt@126.com

one-dimensional convolution neural network [2]. In order to improve the reliability and continuity of the distribution system, Kong et al. built an analysis and monitoring system for the power Internet of Things, and used the system for comprehensive applications such as data acquisition and fault analysis. A fault location method considering distribution phasor measurement unit information and network symmetry was proposed [3]. Machine vision has significantly improved the efficiency, quality and reliability of fault detection. In visual inspection, excellent optical lighting platform and appropriate image acquisition hardware are prerequisites for obtaining high-quality images. Image processing and analysis is the key technology used to obtain fault information; also, deep learning has a significant impact on the field of image analysis.

The application of deep learning plays an increasingly important role in the further development of visual inspection. Therefore, after discussing the traditional defect-detection algorithm, the application of deep learning for defect classification, location and segmentation is described in detail. Ren et al. discussed the brief history and latest technologies of optical lighting, image acquisition, image processing and image analysis in the field of visual inspection. The researcher explained the latest development of industrial fault detection based on machine vision. The development prospects of visual inspection technology were also examined [4]. The wide application of the machine vision system encouraged the development of image-based statistical process control methods to improve the utilization of image data. There has been limited research on fault detection and identification as previous studies have focused on identifying the size and location of faults or detecting the occurrence of faults. Zuo et al. proposed a control mapping method based on regional growth to monitor the image quality of industrial products characterized by uniformity or specific patterns. Simulation results showed that this method not only detects faults quickly, but can also accurately estimate the size and location of faults [5]. However, the researchers did not test the computer software, let alone analyze its fault location.

In order to solve the problem of accurate and fast software fault location, this paper proposes a computer software fault location method based on machine vision and image processing. First, the machine vision technology was studied, and a detailed analysis was conducted of the graph mining technology, and then the practical significance of software fault location was determined based on the current situation. Finally, via an experiment, this paper compared serial and parallel reduction methods. The experimental results indicated that the effect of serial reduction was better when the number of nodes was small, but with the increase of the number of nodes, the fault location time of parallel reduction was shorter and the effect was better. The innovation of this paper was that it not only analyzed the serial-parallel effect of a system, but also carried out secondary analysis in the experimental part, which further confirmed the results obtained from the experiment.

2. FAULT LOCATION ALGORITHM BASED ON VISUAL IMAGE PROCESSING SOFTWARE

2.1 Importance of Software Fault Location

A survey conducted by the National Standards and Technology Association of the United States Department of Commerce shows that the United States suffers losses amounting to billions of dollars each year due to software problems. One can only imagine how serious the consequences would be when the hidden problems in the software are exposed. At the same time, according to statistics, the cost of repairing software defects can account for a large part of the total software maintenance cost, and the maintenance cost itself can account for 40% to 50% of the total software life cycle cost. Therefore, in the early stage of software development, programmers need to adjust the software testing, which can solve the problem as soon as possible and reduce the maintenance costs. If the problem is found late or the repairs take a long time, the costs increase. Errors may occur during the program design and requirement analysis. Because there is a connection and mutual inheritance relationship between software life cycles, the defects of the former would have a greater impact on the later stage, leading to more and more serious problems. With time, the cost of maintenance increases. Assuming that the cost of repairing a failure is 1 in the demand analysis stage, the repair cost is 3 to 6 in the design stage, 10 in the programming stage, 20 to 40 in the internal test stage and 30 to 70 in the external test stage. The maintenance cost of software faults increases with the increase of software life.

From the above discussion, it can be seen that it is very important to recognize the maintenance cost, analyze the cause of the failure, and prevent, find and repair the failure as early as possible to reduce the cost of software [6–7]. However, in practice, people often neglect some inherent characteristics of faults and underestimate, abstract and simplify faults, which is the approach taken by many scholars when locating faults. Therefore, when developing software, the in-depth analysis of various types of faults can effectively find such faults. It is crucial to develop software fault prevention with high security, high availability and high maintainability. In order to prevent faults from occurring or not being identified, this paper re-examines the nature and scope of the problem, and comprehensively identifies the possible problems, thereby playing an important role in improving the efficiency and accuracy of software fault identification.

2.2 Image Processing in Machine Vision System

(1) Image filtering

During their acquisition, images are often contaminated by random signals called ‘noise’. Noise also has a great impact on image quality, and Gaussian white noise and impulse noise are the most commonly occurring types. Gaussian white noise refers to the electronic interference noise generated by the camera, and its brightness follows a Gaussian or normal distribution. On the other hand, impulse noise contains only random positive and negative pulses. Image filtering (also known as local preprocessing) is a technology applied to eliminate noise to a certain extent. Image smoothing technology uses the redundancy of the image to suppress the image, thus removing impulse noise. Median filtering and edge preserving filtering have nonlinear smoothing capabilities. Median filtering is the gray level of each pixel in the window centered on the current pixel, and replaces its middle value with the gray level of the current pixel. This method can better suppress the salt and pepper noise of the pulse interference level, can better suppress the random noise, and can better prevent the edge blur. Median filtering is a typical means of smoothing noise, and can effectively maintain the edge information of the image.

(2) Threshold segmentation

In the actual manufacturing of the placement machine, due to the different shapes of components, the identification methods are also different. However, in order to better recognize the image of the component, it must be separated from the background. This method is called 'image segmentation', for which threshold segmentation (that is, selecting appropriate gray threshold to distinguish the recognized object and background) is the simplest and most effective method. This method can obtain better segmentation results particularly when the gray level of the target object is the same as the same gray level. When selecting the threshold, many methods can be used to divide it into global threshold, local threshold and adaptive threshold. For the overall threshold, a unified threshold is used to binarize the entire image. The adaptive threshold method uses a series of thresholds related to coordinates to separate each image (that is, the relationship between threshold and coordinates). The local threshold method is a general global threshold method.

2.3 Fault Location of Image Processing Software

The locating of a computer software fault is an important part of computer fault diagnosis, and is also a complex and difficult task. It requires not searching in the database according to the diagnostic parameters, but using the image processing function of the computer to quickly and accurately determine the problem through the analysis of the execution result image when the program is running [8–9]. The advantages of a machine vision system are fully exploited when locating a computer software fault. Firstly, the machine vision technology collects the images generated during program operation and transmits them to the computer [10–11]. Then, when the computer software is operating, the execution results are obtained by image processing of the program execution results. Finally, through the diagnosis module of the computer software fault location software system, the image features of the execution results are extracted as the diagnosis parameters, and analyzed [12–13].

(1) Graph mining technology

This paper proposes a software fault location technology based on graph mining technology, uses a new method, and describes its characteristics. It also carries out a differential operation on it, determines the probability ranking of fault occurrence, and improves the efficiency and accuracy of detection [14–15]. At present, using graph mining technology to locate software faults has become a hot but difficult issue in the current software field. Hence, system innovation and improvement would greatly reduce the cost of software testing, thus ensuring the safety and reliability of application software in various fields.

After obtaining the target image, in order to better meet the subsequent image mining work, image processing and object recognition processes are conducted. For example, image noise reduction is performed on the image affected by noise. Image enhancement is performed on the image with weak information or the image object concerned by the user. The distorted image is used for geometric correction and the restoration of a degraded image.

Unlike the structural data in the database, the image data is a type of unstructured or semi-structured data. If the traditional data mining technique is used, then before mining out images, a series of processing stps must be undertaken. Similar to knowledge discovery and the data mining of databases, image mining is also a multi-stage process that requires a great deal of backtracking. During this process, each preceding and following step can be analyzed.

(2) Grayscale histogram

The grayscale histogram of a digital image is essentially a one-dimensional discrete function of a one-dimensional digital image:

$$g(k_r) = m_r \quad r = 0, 1, \ldots, Z-1 \tag{1}$$

where k_r is the r-level gray level of the digital image; m_r is the number of pixels in the gray level k_r of the digital image; Z is the number of gray scales displayed in the digital image.

In fact, the histogram of digital image standardization is usually used, and its normalized histogram $q(k_r)$ is obtained by the following formula:

$$q(k_r) = m_r/m \quad r = 0, 1, \ldots, Z-1 \tag{2}$$

where m represents the total number of pixels in the digital image. In fact, $q(k_r)$ gives the possibility of gray scale in digital images, and it is not difficult to find:

$$\sum_{r=0}^{Z-1} q(k_r) = 1 \tag{3}$$

Through the definition of standard histogram, other statistical parameters can also be obtained. If k is a discrete random variable on the interval 0, $Z-1$, and $q(k_j)$ represents the standardized histogram component of the j-th finger corresponding to k, then the mean value n of k is the mean value of gray scale.

$$n = \sum_{j=0}^{Z-1} k_j q(k_j) \tag{4}$$

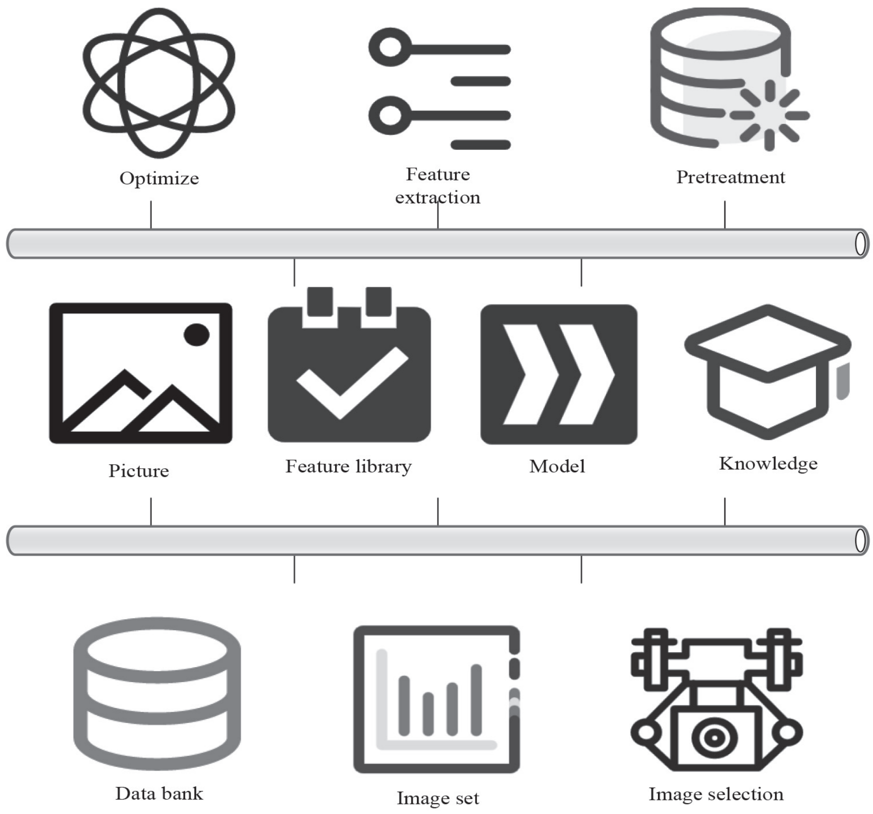


Figure 1 Image mining framework model.

The variance of k can be defined as:

$$\beta^2(k) = \sum_{j=0}^{Z-1} (k_j - n)^2 q(k_j) \tag{5}$$

The gray histogram is an important index in image processing, analysis, understanding and image mining.

(3) Graph reduction

Graph reduction is a key step of software fault diagnosis based on graph mining technology. In order to improve the positioning efficiency of the system, the graph reduction technology is used to simplify the program call graph extracted from the wrong program, and the call graph is used to represent the running path of the program. However, the current graph mining algorithm has great limitations, so it must be simplified. The simplification process is usually a lossy information compression process. Its main function is to minimize the size of the image while maintaining the information of the original image as much as possible.

(4) Characteristic information analysis

The feature information analysis and difference operation involves combining the obtained feature information with the unqualified test results through the mining method of analysis graph to obtain the corresponding feature information. An error does not have any impact on the same method calls, but does have an impact on the same method calls in the same environment. After the feature information table of frequent subgraphs is generated, the skepticism formula can be used for analysis and calculation. The formulas are:

$$Info(T) = -\sum_{j \in T_E} \frac{|T_{E=j}|}{|T|} \cdot \log_2 \left(\frac{|T_{E=j}|}{|T|} \right) \tag{6}$$

$$InfoGain\ (X, T) = Info(T) - \sum_{i \in T_X} \frac{|T_{X=i}|}{|T|} \cdot Info(T_{X=i}) \tag{7}$$

where T is the calculated data, and E is any column in the data.

Because the number of successful executions of method calls represented by each column is fixed, $Info(T)$ is a fixed value, and $InfoGain(X, T)$ calculates the contribution of edges belonging to X-th column to the error. The larger the value, the higher the position in the suspicious sorting.

Because of the importance of software fault location research, countless fault location technologies have emerged. How to accurately locate and evaluate faults has become an urgent problem to be solved. According to the characteristics of this kind of fault location system, some scholars propose evaluation indicators such as location time requirement, operation space requirement, test case requirement and location score, among which the location accuracy of location results is the most important evaluation indicator [16–17].

3. COMPUTER SOFTWARE FAULT LOCATION METHOD

3.1 Simulation Experiment

(1) Experimental environment

In order to confirm that the parallel subtree reduction method can efficiently improve the effectiveness of fault location in software fault detection, this paper proposes a

Table 1 Test platform environment.

Category	Configuration items	Information
Hardware	Model	Pentium(r)Dual-Core
	Central Processing Unit (CPU)	T4300
	Random Access Memory (RAM)	2GB
	Hard Disk Drive (HDD)	320GB
Software	Operating System:	Windows 7 Professional
	Development Environment	Eclipse IDE for Java Developers 4.3.0
	Programming Language	Java 8 (JDK 1.8.0_281)

parallel subtree reduction algorithm to solve the problems existing in the commonly used graph-based software fault location technology. The Fork or Join parallel algorithm was used to decompose the target problem into multiple subtasks through repeated iteration and scribing operations, thus effectively simplifying the call graph; The experiment was compiled and implemented in Java using the Eclipse 4.3.0 integrated development environment. The complete test platform environment configuration is shown in Table 1.

(2) Applying AspectJ framework to obtain program call graph

The program call graph was obtained when executing the wrong program. It was an input of graph mining technology. The traditional program running path was obtained by insertion, which was obtained by artificially adding code. Plug-in was a simple and fast solution but, in real life, the project scale of the system was often large, required a lot of labor, and was not convenient for later modification and maintenance. To address the shortcomings of traditional interpolation algorithms, this paper proposes an Aspect-Oriented Programming (AOP) approach based on AspectJ, which is simple and practical. The AspectJ framework extends Java, and its new syntax structure makes modular implementation of aspect-oriented programming possible.

A connection point defines a specific node of a program. For example, if a method is a connection point, its class and object initialization can be called a connection point. Its function is to help programmers track its position in the execution process.

Notification is a kind of code used when defining connection points during program operation. It can be divided into many types according to the needs of users. The most basic is that before and after connection points are run, they are known as pre-notification and post-notification.

Pointcut: the correspondence between notification and connection points can be one-to-many. People call this group of connection points “pointcuts”. Its function is to enable programmers to control the corresponding code, thus achieving the corresponding notification.

Aspect: this is the total number of notifications and pointcuts. Therefore, Aspect not only controls the execution of notifications, but also defines different logical relationships in pointcuts.

Weaving: this occurs when an aspect of the execution program is added to the code.

The program call graph obtained in this paper is the execution trace graph in the error program; it defines the program operation mode as the connection point, and uses a notification to implement all methods. That is, it selects all methods from the program. During program operation, every time the error program runs to the connection point defined by AspectJ, the program in this notification begins, records the call information of the current method, including method name, method caller, callee, and so on, and saves it in the form of a chart. The information recorded in the chart is the wrong program called by a program, that is, the running track of a method.

When using graph mining technology to locate software faults, a frequent subgraph is used. That is to say, the program call curve to be obtained is not a single run of a program. It is used to test the wrong program several times, capture the track of a program call during each test, and determine whether the test is successful or has failed according to the consistency of the output results and expectations. The specific steps are shown in Figure 2. In the software fault location technology based on graph mining technology, the contribution of the fault occurrence method to the fault is determined through comparative analysis and calculation, and the result of ranking from large to small is given.

(3) Experimental data

As well as dealing successfully with common faults, the reduction problem of traditional subtree and parallel subtree is definitely solved. The reduction process and results calculated for the two methods are compared, thus proving the effectiveness of parallel subtree reduction. This paper takes the actual data and the simulation data as the test data to meet the test requirements and ensure the accuracy of the test results.

The actual data was obtained from the open-source test software, Nanoxml analyzer. Because the reduction time is often closely related to the number of nodes in the simplified graph, six different Nanoxml functions were designed in this paper. There are six different calculation methods. The number of each calculation method is different, ranging from hundreds to tens of thousands. It can be a good comparison between traditional subtree and parallel subtree simplified algorithms. In this experiment, the number of six nodes of Nanoxml is 232, 1648, 5646, 6274, 6666 and 7421.

In addition, in order to verify that the reduction efficiency of the proposed image processing algorithm is higher when the number of nodes in a certain range is larger, the number of layers and methods were calculated by mens of a loop so as to obtain different numbers of nodes. The parallel subtree reduction method algorithm was used to reduce the corresponding program under test in a serial way, and the times

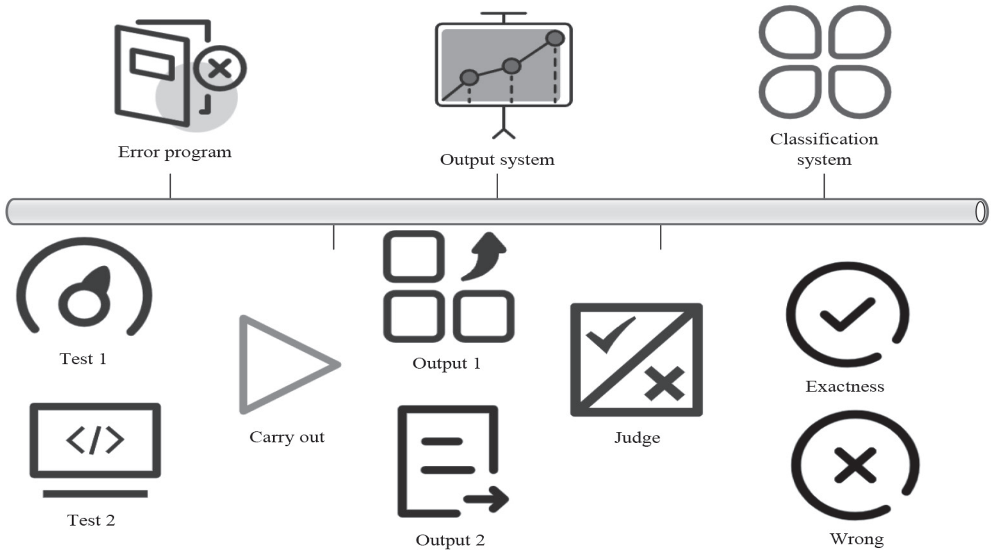


Figure 2 Trace judgment program diagram of program call.

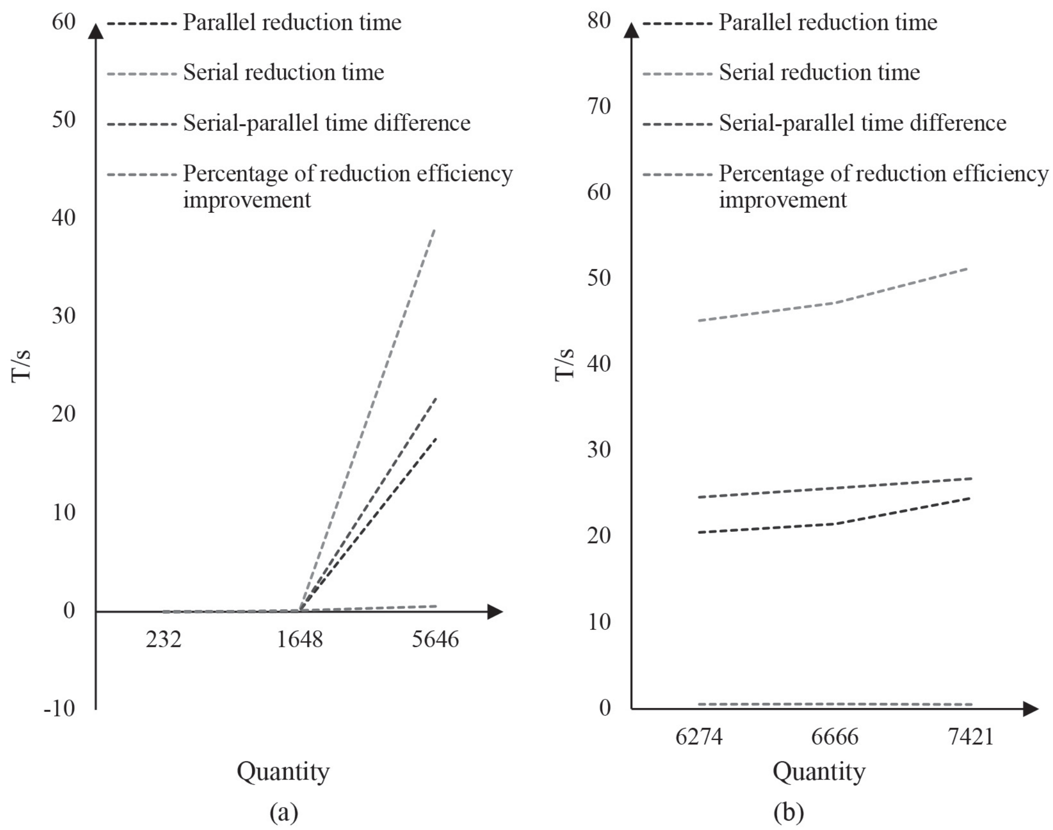


Figure 3 Serial-parallel reduction time data diagram of Nanoxml test program.

obtained from the testing of the parallel subtree reduction algorithm were compared. Eight sets of experimental data were used, comprising 866, 2476, 5438, 8422, 12786, 14789, 17632 and 26824 nodes respectively.

3.2 Experimental Evaluation

(1) Test results and analysis of Nanoxml program

For the experiment, this paper first took six programs with different functions of the Nanoxml analyzer as the test object, then used the traditional serial subtree reduction algorithm and the parallel subtree reduction algorithm of the Fork or Join framework to test the five programs. The the time required for the simplification process under the two algorithms was recorded. In this paper, the traditional subtree and parallel subtree reduction algorithm were used. The reduction time of each node is shown in Figure 3. The serial reduction was performed for six Nanoxml parsers with different number of nodes and six programs with different number of nodes. The recorded times are shown below. Because the time required

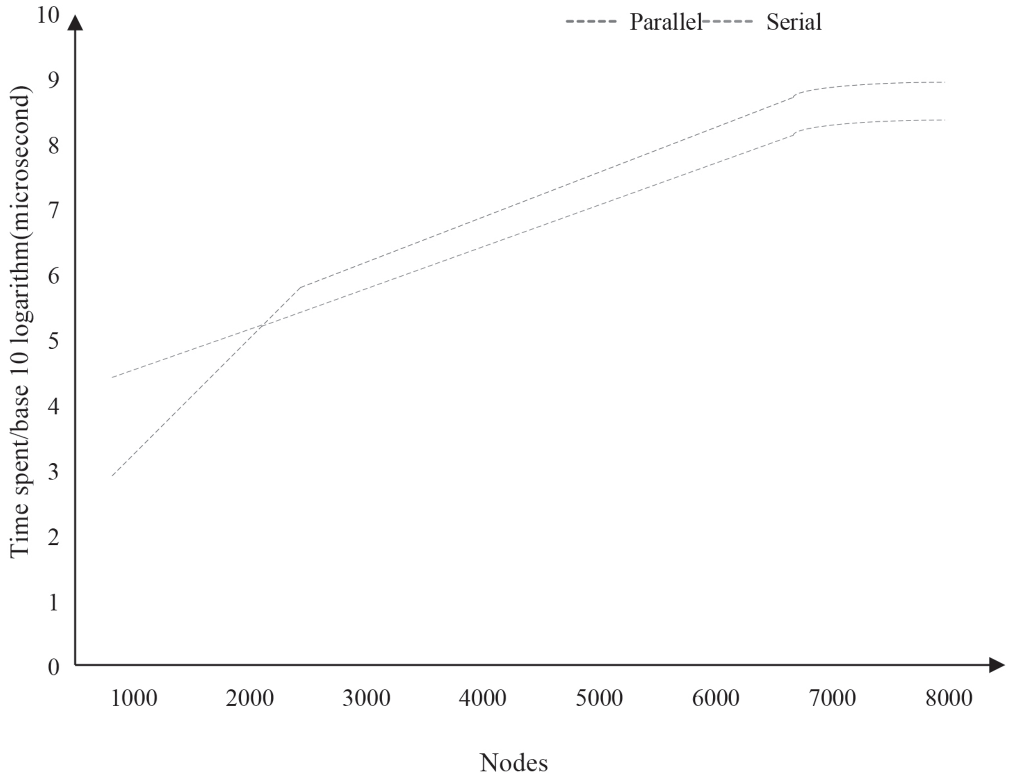


Figure 4 Diagram comparing reduction time of serial and parallel connections of Nanoxml test program.

to simplify the program with a small number of nodes was very short, it was taken as the time unit.

As shown in Figure 3, Figure 3 (a) shows the serial-parallel reduction time data of the three node-number Nanoxml test programs of 232, 1648 and 5646, and Figure 3 (b) shows the serial-parallel reduction time data of the three node-number Nanoxml test programs of 6274, 6666 and 7421. It can be seen from the figure that when the number of nodes is small, the serial reduction still had some advantages. However, when there is a greater number of nodes, the advantages of parallel reduction become more obvious. When the number of nodes is 5646, the efficiency improvement value reaches 55.22%.

Based on the test data in Figure 3, the time comparison diagram of the concatenation and reduction of the Nanoxml test program shown in Figure 4 was drawn. Its abscissa represents the number of nodes of the program being tested. It pertains to the program call graph obtained in the error program, that is, the number of nodes contained in the original graph, rather than the number of methods offered by the program being tested. Because the simplification time required by the test procedure for the number of nodes at each stage was very different, in order to facilitate the diagram, its ordinate was expressed in microseconds as a logarithm with a base of 10.

By reducing series-parallel subtrees for six different test methods, the corresponding time data were obtained. From Figure 4, it can be seen that parallel subtree reduction in small node test cases has no advantage in terms of efficiency as it takes longer. When the number of nodes exceeds 2000, the calculation result of parallel subtree reduction is more efficient.

(2) Test results and analysis of simulation program

According to the different functions of Nanoxml, the number of nodes in the test program was limited. At the same time, in order to verify the advantages of the parallel subtree reduction algorithm in the program call graph with a large number of nodes, a set of data sets were tested serially using the self-designed test software, with the number of nodes being 866, 2476, 5438, 8422, 12786, 14789, 17632 and 26824. The results are shown in Figure 5.

As shown in Figure 5 (a), the serial and parallel reduction time data diagram of the simulation program with the number of 866, 2476, 5438 and 8422 nodes, and Figure 5 (b) shows the serial and parallel reduction time data diagram of the simulation program for 12786, 14789, 17632 and 26824 nodes. It can be seen from the figure that there is still no obvious advantage in time in the summary test case. Only when the number of nodes is more than 866 and less than 2476, does the parallel ability begin to appear. With the increase of the number of nodes, its advantage in terms of time becomes increasingly obvious. When the number of nodes exceeds 25000, the time required for serial execution is about 15 minutes, while the time required for parallel processing is only about 8 minutes, greatly improving the efficiency of software fault location.

4. CONCLUSIONS

This paper introduced a parallel subtree reduction algorithm based on verification research. Its experimental design and result analysis showed that this algorithm was superior to the traditional serial subtree reduction algorithm in improving the reduction speed. This paper used Nanoxml software

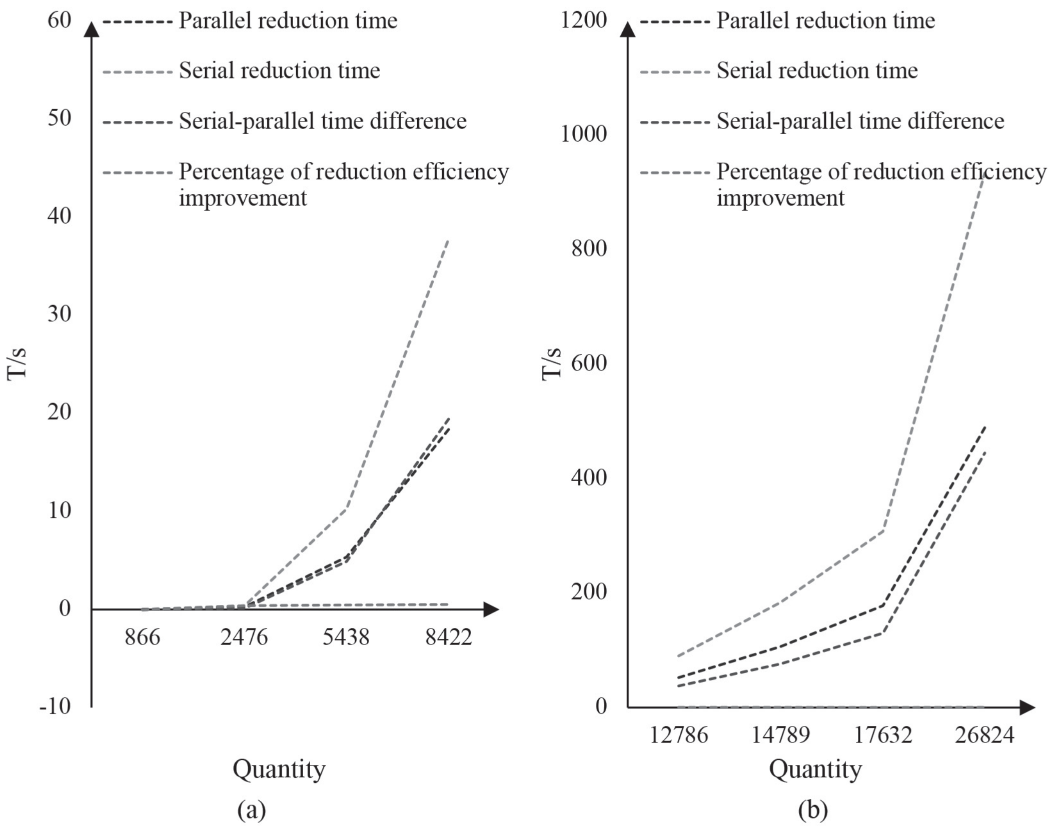


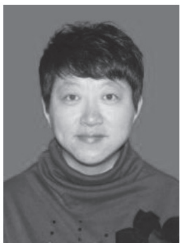
Figure 5 Serial-parallel reduction time data diagram of simulation program.

and a simulation program to test programs with different node numbers, record them and compare them. Through the analysis of an example, it was proved that the parallel subtree reduction algorithm can effectively reduce the number of nodes in the graph, and the efficiency of graph reduction can reach 55.22%, greatly improving the efficiency of fault location. Moreover, this paper elaborated the basic principle of call graph under the AspectJ framework, emphasized the flexibility of this method, and improved the efficiency of the reduction process to a certain extent.

REFERENCES

1. Naidu O. D. and Ashok Kumar Pradhan. "A traveling wave-based fault location method using unsynchronized current measurements". IEEE Transactions on Power Delivery, 34.2 (2018): 505–513.
2. Lan Sheng, Mou-Jie Chen and Duan-Yu Chen. "A novel HVDC double-terminal non-synchronous fault location method based on convolutional neural network". IEEE Transactions on Power Delivery, 34.3 (2019): 848–857.
3. Kong Xiangyu, Yong Xu, Zaibin Jiao, Delong Dong, Xiaoxiao Yuan, Shupeng Li. "Fault location technology for power system based on information about the power internet of things". IEEE Transactions on Industrial Informatics, 16.10 (2019): 6682–6692.
4. Ren Zhonghe, Fengzhou Fang, Ning Yan, You Wu. "State of the art in defect detection based on machine vision". International Journal of Precision Engineering and Manufacturing-Green Technology, 9.2 (2022): 661–691.
5. Zuo Ling, Zhen He and Min Zhang. "An EWMA and region growing based control chart for monitoring image data". Quality Technology & Quantitative Management, 17.4 (2020): 470–485.
6. Huang, Z., Chen, Y., Liu, J., Ni, H., & Zhang, D. (2023). Rolling bearing fault diagnosis based on the enhanced channel attention network. Cyber-Physical Systems, 10(2), 197–213.
7. Ghorbani Amir and Hasan Mehrjerdi. "Negative-sequence network-based fault location scheme for double-circuit multi-terminal transmission lines". IEEE Transactions on Power Delivery, 34.3 (2019): 1109–1117.
8. Yang Hui, Xudong Zhao, Qiuyan Yao, Ao Yu, Jie Zhang, Yuefeng Ji. "Accurate fault location using deep neural evolution network in cloud data center interconnection". IEEE Transactions on Cloud Computing, 10.2 (2020): 1402–1412.
9. Usman Muhammad Usama and Md Omar Faruque. "Validation of a PMU-based fault location identification method for smart distribution network with photovoltaics using real-time data". IET Generation, Transmission & Distribution, 12.21 (2018): 5824–5833.
10. Lin Tzu-Chiao and Zong-Jhen Ye. "A signal-superimposed technique for fault location in transmission lines through IED measurements considering communication service failure". IEEE Systems Journal, 15.3 (2020): 4525–4536.
11. Aminirad Mohammad and S. Mohammad Shahrtash. "A novel protection and fault location combined algorithm for mixed/hybrid lines". IET Generation, Transmission & Distribution, 16.10 (2022): 2016–2031.
12. Mirzaei Mahdi, Behrooz Vahidi and Seyed Hossein Hosseinian. "Fault location on a series-compensated three-terminal transmission line using deep neural networks". IET Science, Measurement & Technology, 12.6 (2018): 746–754.

13. Gururajapathy Sophi Shilpa, Hazlie Mokhlis and Hazlee Azil Illias. “Fault location using mathematical analysis and database approach”. COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 38.1 (2019): 415–430.
14. Gupta Neha, Arun Sharma and Manoj Kumar Pachariya. “Multi-objective test suite optimization for detection and localization of software faults”. Journal of King Saud University-Computer and Information Sciences, 34.6 (2022): 2897–2909.
15. Liu Keyan, Tianyuan Kang, Xueshun Ye, Muke Bai, Yaqian Fan. “A fault location method of distribution network based on XGBoost and SVM algorithm”. IET Cyber-Physical Systems: Theory & Applications, 6.4 (2021): 254–264.
16. Zhanyong Wu, Xiaohang Ma, Yanxue Li. “Optimizing Quantitative Intelligent Systems Investment Strategies and Asset Allocation with Machine Learning.” International Journal of Engineering Intelligent Systems, 33.4 (2025).
17. Xiaojun Zhang. “Deep Learning Model for Image Classification in Machine Vision.” International Journal of Engineering Intelligent Systems, 33.4 (2025).



Wei Yan was born in Qiqihaer, Heilongjiang Province, China, in December, 1980. She is an associate professor at Liaoning Police College. She received the bachelor’s degree from Qiqihaer University, her master’s degree from Harbin University of Science and Technology. Her research interests include computer vision and image recognition.

E-mail: yanweiln@126.com

Chen Yang was born in Harbin, Heilongjiang Province, China, in 1980. He is an associate professor at Dalian University of Foreign Languages. He received the bachelor’s degree from Qiqihar University, his master’s degree from Harbin University of Science and Technology. His research interests include natural language processing, federal learning and privacy protection.

E-mail: yangchenyjt@126.com