

# A Method to Identify Critical Operations in Accelerated Software Reliability Testing

Qiuying Li<sup>1</sup> and Lei Luo

School of Reliability and Systems Engineering, Beihang University  
Beijing, China

**Abstract.** The foundation of the accelerated software reliability testing (ASRT) based on importance sampling is to construct strengthened operational profile (SOP). The key of SOP's construction is to identify critical operations properly which have dominant effects on software's reliability. In this paper, a method based on Analytic Hierarchy Process (AHP) method is proposed to identify the critical operations. The hierarchies of goals, criteria and alternative programs have been defined separately. Finally a preliminary application is practised to show the method's application process and demonstrate the feasibility of this method, which can be taken as a reference for critical operations's identification when using ASRT in engineering application.

**Keywords:** critical operations, software reliability, accelerated software testing, analytic hierarchy process.

## 1. Introduction

For high reliability software, traditional SRT method faces a problem that few or zero failure appears after a long period of execution time for software reliability testing, or when the reliability index is very high (e.g. the failure rate is up to  $10^{-7}/h$ ) the required test duration time is up to thousands of years. All above is obviously unacceptable in engineering practice [1]. In order to solve this problem, importance sampling (IS) method is applied into SRT [2], [3]. Reference [4] converts traditional Musa's operational profile into SOP, whose operations have dominant effects on software's reliability. Those defects founded during these operations' running have great impact on high reliability software to achieve the high reliability goal in a shorter period time, i.e. accelerating the SRT's duration.

The foundation of such ASRT method is to construct SOP reasonably and effectively [5]. The key of SOP's construction is to identify critical operations properly, then construct SOP by redistributing these operations' occurrence probability. Whether the critical operations can be identified correctly is related to accelerate or not and acceleration efficiency.

Reference [3] puts forward that for high reliability software, failure is more likely to occur in those operations of important functions, such as the redundant protection, exception handling and the initial calibration operations. According to existing data, [6] concludes experience as guidance of critical operations identification but only can be referred by some similar software. Reference [7] studies on software safety testing using fault tree analysis (FTA) method to identify hazard operations. Through FTA to find the basic events which may cause the incident, find out operations related to these events, then construct a correlative risk profile [8]. This approach is considerable, but for high reliability software risk of accidents is only part of factors should be considered. AHP [9] is a multi-criteria decision-making technique that has been widely used to solve complex decision problems [10], in which both qualitative and quantitative aspects are considered.

---

<sup>1</sup> Corresponding author. Tel.: + 8601082339169; fax: +8601082313913.  
E-mail address: li\_qiuying@buaa.edu.cn.

Moreover AHP has been widely applied in engineering, industry, manufacturing and so on<sup>[11]</sup>. This paper puts forward a new method which uses AHP to identify critical operations, and gives a systematic and comprehensive guidance to application.

## 2. Identify the Critical Operations

Now there exist some methods to identify critical operations, but the problem is that whose analysis is not comprehensive enough. It may miss some of very important critical operations.

After Musa's operational profile is constructed, the basic characteristics of each operation has been identified clearly. However, the influencing degree of each operation's execution for high reliability software is uncertain. It is related to the probability of occurrence, the consequences of failure and other factors. These factors have different weights, therefore a comparative analysis of all factors have some difficulty. The introduction of AHP can help to convert complex problems into some orderly hierarchies and a variety of factors. AHP is a way to analyze quantitative problem qualitatively and provide a simple approach for decision-making.

### 2.1. Analytical Hierarchy Process

The basic principle of AHP is to convert complex problem into several levels and factors. Then every two factors contrast with each other to get the weights separately. From analysis and calculation through low to high layers, each program's weight is obtained for the overall goal. The largest weight of the program is the optimal solution<sup>[11]</sup>.

### 2.2. The Procedure to Identify Critical Operations

- Construct a multi-level hierarchical structure

According to the characteristics of high reliability software, to identify the critical operations in Musa's OP is based on the following four evaluation criteria: i)Operation's probability; ii)Mission-critical level; iii)Consequence of operation's failure; iv)Input domain of operation. Then a hierarchical model can be built as shown in Fig. 1.

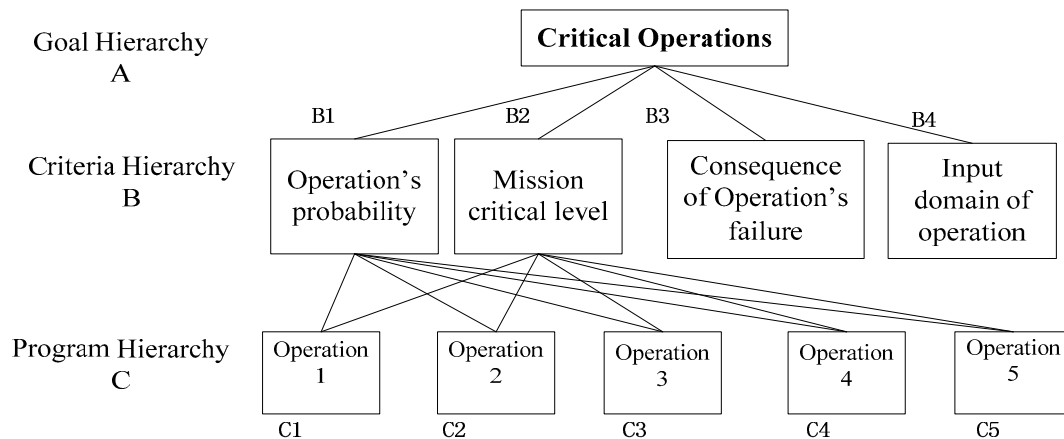


Fig. 1 Hierarchical model

- Construct the comparison matrix of criteria hierarchy

Table 1 The definition of 1-9 Scale

Relative importance $a_{ij}$	Definition
1	The goal $i$ and $j$ are equally important
3	The goal $i$ is a little more important than $j$
5	The goal $i$ is more important than $j$
7	The goal $i$ is much more important than $j$
9	The goal $i$ is more important than $j$ absolutely
2, 4, 6, 8	Intermediate levels of strength

Matrix that takes the two programs (or sub-goals) of the relative importance as elements is called comparison matrix. 1-9 scale is used to determine the matrix. Table 1 shows the meaning of the scale. After analyzing, the comparison matrix of criteria hierarchy is given as follows:

$$\begin{array}{c|cccc}
 A & B1 & B2 & B3 & B4 \\
 \hline
 B1 & 1 & 3 & 5 & 7 \\
 B2 & 1/3 & 1 & 1/2 & 4 \\
 B3 & 1/5 & 2 & 1 & 5 \\
 B4 & 1/7 & 1/4 & 1/5 & 1
 \end{array} \tag{1}$$

- Construct the comparison matrix of operation hierarchy based on every criterion

There are four criteria to identify critical operations. If  $n$  operations are to be decided, four  $n \times n$  matrixes (F) should be built, representing the importance weights of all operations based on these four criteria.

- Normalize the comparison matrix and calculate the weights

Firstly, use normalization method to calculate each matrix line, then convert ( $F$ ) to ( $F'$ ). Secondly, the sum of each matrix row should be calculated. At last, determine weights for all the row sum. Here take comparison matrix of criteria hierarchy for example shown in Table 2.

Table 2 Weights of the criteria hierarchy

A	B1	B2	B3	B4	Row sum	Weight
B1	0.5967	0.48	0.7463	0.4118	2.2348	0.5587
B2	0.1989	0.16	0.0746	0.2353	0.6688	0.1672
B3	0.1193	0.32	0.1493	0.2941	0.8827	0.220675
B4	0.0851	0.04	0.0298	0.0588	0.2137	0.053425
Sum	1	1	1	1	4	1

- Consistency analysis

The elements of comparison matrix have three mathematical properties.  $a_{ii} = 1$ ,  $a_{ij} = 1/a_{ji}$ ,  $a_{ij} = a_{ik} \cdot a_{kj}$ . Comparison matrix is generally estimated based on knowledge and experience of policy makers. As decision-makers do not have accurate estimates, they may fail to meet the need of the third property. Since consistency analysis is necessary, one should calculate the average random consistency ratio  $CR$ .

When  $CR < 0.1$ , it means that comparison matrix conforms to consistency. Random consistency index  $RI$  is available by look-up table. Then we have

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{2}$$

where

$$\lambda_{\max} = \frac{1}{n} \sum \frac{(FW)_i}{w_i} \tag{3}$$

where  $F$  is the comparison matrix,  $W$  is the weights vector.

Then we have

$$CR = \frac{CI}{RI} \tag{4}$$

- The final selection results

Weights vector of criteria hierarchy is given as follows.

$$B_{41} = [0.5587 \quad 0.1672 \quad 0.220675 \quad 0.053425]^T \tag{5}$$

Then we have

$$V_{n1} = [C_{ij}]_{n4} \cdot B_{41} \tag{6}$$

We get the last matrix  $V_{41}$  as shown in Table 3, which represents total weights of all the operations. Sort the operations by total weights, we can choose the front one-third or half operations as critical operations, according to the accelerated degree we want.

Table 3 Calculation of the total weights

Criteria	Operation's probability	Mission-critical level	Consequences of operation's failure	Input domain of operation	Total weights of all the operations
Weights of the criteria hierarchy	0.5587	0.1672	0.220675	0.053425	$V_{n1}$
Weights of Operation 1					
programs Operation 2					
hierarchy for the ..... four Criteria Operation n					

### 3. Example

Here we take a flight control software for example. As the actual OP is complicated, here we analyze only a small part of operations in the Musa's OP and aim to show the process to identify critical operations.

- List the operations with their probability as programs hierarchy.

Operations should be analyzed are: C1: Self-test (0.021); C2: Data-loading (0.057); C3: Aerial cannon attack (0.000134); C4: Missile attack (0.000089); C5 Patrol (0.048).

- If no any other criterion to be added, the weights of criteria hierarchy can use the values of  $B_{41}$  directly.

$$B_{41} = [0.5587 \quad 0.1672 \quad 0.220675 \quad 0.053425]^T.$$

- Construct the comparison matrix of operations hierarchy based on each criterion.

The comparison matrixes for the four criteria are shown as follows.

$B1$	C1	C2	C3	C4	C5	$B2$	C1	C2	C3	C4	C5	$B3$	C1	C2	C3	C4	C5	$B4$	C1	C2	C3	C4	C5	
$F_1 =$	C1	1	1/3	1/6	1/8	1/2	C1	1	2	1/7	1/8	2	C1	1	5	3	3	5	C1	1	1/7	1/3	1/3	1/2
	C2	3	1	1/5	1/6	2	C2	1/2	1	1/7	1/8	1	C2	1/5	1	5	5	6	C2	7	1	5	5	6
	C3	6	5	1	1/2	7	C3	7	7	1	1	7	C3	1/3	1/5	1	1	7	C3	3	1/5	1	1/2	2
	C4	8	6	2	1	7	C4	8	8	1	1	7	C4	1/3	1/5	1	1	7	C4	3	1/5	2	1	5
	C5	2	1/2	1/7	1/7	1	C5	1/2	1	1/7	1/7	1	C5	1/5	1/6	1/7	1/7	1	C5	2	1/6	1/2	1/5	1

- Calculate the weights vector.

The weights vector of four comparison matrixes are shown as follows.

$$W_1 = \begin{bmatrix} 0.04334 \\ 0.097099 \\ 0.326584 \\ 0.470828 \\ 0.062149 \end{bmatrix} \quad W_2 = \begin{bmatrix} 0.077261 \\ 0.049741 \\ 0.39973 \\ 0.422021 \\ 0.051246 \end{bmatrix} \quad W_3 = \begin{bmatrix} 0.405852 \\ 0.293151 \\ 0.131609 \\ 0.131609 \\ 0.037779 \end{bmatrix} \quad W_4 = \begin{bmatrix} 0.053084 \\ 0.542616 \\ 0.12515 \\ 0.203579 \\ 0.075239 \end{bmatrix}$$

- Consistency analysis

For the first matrix, we have:

$$\lambda_{1\max} = \frac{1}{5} \left( \frac{0.2201}{0.04334} + \frac{0.4954}{0.097099} + \frac{1.7426}{0.326584} + \frac{2.4881}{0.470828} + \frac{0.3114}{0.062149} \right) = 5.162.$$

As  $n$  is 5, we have  $RI=1.11$ ,  $CI_1 = \frac{5.162-5}{5-1} = 0.04$ ,  $CR_1 = \frac{0.04}{1.11} = 0.036 < 0.1$ . We know matrix  $F_1$  has good consistency. In the same way we get  $CR_2 = 0.0176$ ,  $CR_3 = 0.023$ ,  $CR_4 = 0.011$ .

- Overall weight

$$V_{51} = \begin{bmatrix} 0.04334 & 0.077261 & 0.405852 & 0.053084 \\ 0.097099 & 0.049741 & 0.293151 & 0.542616 \\ 0.326584 & 0.39973 & 0.131609 & 0.12515 \\ 0.470828 & 0.422021 & 0.131609 & 0.203579 \\ 0.062149 & 0.051246 & 0.037779 & 0.075239 \end{bmatrix} \bullet \begin{bmatrix} 0.5587 \\ 0.1672 \\ 0.220675 \\ 0.053425 \end{bmatrix} = \begin{bmatrix} 0.12953 \\ 0.156246 \\ 0.285026 \\ 0.373533 \\ 0.055648 \end{bmatrix}$$

According to  $V_{51}$  the weights order of the programs hierarchy is:  $C4$ :Missile attack,  $C3$ :Aerial cannon attack,  $C2$ :Data-loading,  $C1$ :Self-test,  $C5$ :Patrol. So the critical operations are  $C4$  and  $C3$ .

## 4. Conclusion

It is very important to identify critical operations properly when using the accelerated software reliability testing method. A method based on AHP is put forward to identify critical operations in this paper. It helps to think more systematically and reduce the risk of wrong decision. All the processes of this method are given and some results are shown by an example. Further validation should be further studied in future work and development of the assistant tool for this method should be further considered.

## 5. References

- [1] R W. Butler and G. B. Finelli. The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software. *IEEE Transactions on Software Engineering*, 1993, 19(1):3-12.
- [2] S. Alam, H. Chen and K. Ehrlich. Assessing software reliability performance under highly critical but infrequent event occurrences. *Proceedings of The Eighth International Symposium On Software Reliability Engineering*, 1997:294-307.
- [3] M. Hecht and H. Hecht. Use of importance sampling and related techniques to measure very high reliability software. *Aerospace Conference Proceedings 2000*, 533-546.
- [4] Q. Y. Li, X. Li, J. Wang and L. Luo. Study on the accelerated software reliability demonstration testing for high reliability software based on strengthened operational profile. *Proceedings of international conference on Computer Technology and Development (ICCTD)*, 2010.
- [5] Z. D. Qin, Y. Cai, H. Y. Wang and X. Q. Liu. Software Safety Demonstration Test Method Based on Accelerating Profile. *Computer Engineering*, 2008, 34(3)(in Chinese).
- [6] D. Tang, M. Hecht, J. Miller, L. Czekalski and J. Handal. MEADep and Its Applications in Evaluating Dependability for Air Traffic Control Systems, *Proceedings of the Annual Reliability and Maintainability Symposium*, Anaheim, CA, Jan. 19-22, 1998.
- [7] Z. D. Qin, X. Q. Liu, H. Y. Wang and J. J. Le. Software safety growth testing method based on correlative risk profile. *Systems Engineering and Electronics*. 2009, 31(5)(in Chinese).
- [8] Z. D. Qin. *Study on Testing and Evaluation Theory of Highly Credible Software*. Ph.D thesis, 2005(in Chinese).
- [9] T. L. Saaty. *Decision Making with Dependence and Feedback*. Analytic Network Process, Pittsburgh: RWS, 2001.
- [10] L. B. Yang and Y. Y. Gao. *Fuzzy mathematic elements and application*. GuangZhou: South China University of Technology Press, 2002.
- [11] O. S. Vaidya and S. Kumar. Analytic hierarchy process: An overview of applications. *European Journal of Operational Research*. 2006, 169(1):1-29.