Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2015, *9*, 199-203 199

Open Access

Research of Component Software Reliability Estimation

Liu Shuchen[1,*] and Cao Wei[2]

[1]*College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, 210094, P.R. China;* [2]*College of Communication, Nanjing University of the Arts, Nanjing, Jiangsu, 210013, P.R. China*

Abstract: With the rapid development of component software technology, it is very important how to improve the single component software reliability and how to estimate the whole software system reliability using component software reliability. In this paper, we firstly analyze the defects of the traditional method to calculate the component software reliability estimation. Then, a new strategy is presented for solving this problem. We call the strategy as variance decomposition. Finally, we verify the advancement of the approach in the theory.

Keywords: Component software, confidence interval, reliability estimation variance, variance decomposition.

1. FOREWORD

Today, with the rapid development of computer science, the software development of the large complicated software system, based on the component software design, is turning mature day by day. Whereas, with the main emphasis put on the development technology of the component software and the reuse technology of the component software, less attention has been attached to the quality problems, like the component software reliability [1].

To improve the reliability of the component software, it is a common practice to calculate or estimate the reliability of the software system. As the available data of the component software rating is rather limited, so the accuracy of the estimation of the reliability, based on the data, would be greatly affected. Besides, when the same performance is required, it is possible to use the same component software in various systems or in different positions of the same system. So it is necessary to take the variance decomposition and confidence interval into definite consideration.

2. TRADITIONAL RELIABILITY ESTIMATION

The pattern of the software reliability estimation can be divided into the reliability distribution pattern and redundancy distribution pattern. For the reliability distribution pattern, the reliability of the component software is regarded as a decisive variable. As for the redundancy distribution pattern, the decisive variable is the choice of the redundance rating and component software.

2.1. The Pattern of Reliability Estimation

Hwang [2] put forward a reliability distribution pattern to maximize the reliability of the complicated system. This pattern defines the generalized Lagrange Function, and provides a constraint condition, as well.

$$\max \; f(r) = R(r_1, r_2, \cdots r_s) \tag{1.1.1}$$

Constraint condition:

$$g_j(r) \le b_j$$
$$0 < r_i \le 1$$

Here: r_i = reliability of the component software i, i = $1, 2, \cdots s$

b_j = constraint limit, j = $1, 2, \cdots m$

Sayama has spread the pattern, and turned it into a non-linear programming with no constraint condition.

Here is the problematic function:

$$L(r,\lambda;\beta) = f(r) - \sum_{j=1}^{m} \lambda_j (g_j(r) - b_j) + \sum_{j=1}^{m} \left\{ \begin{array}{l} \beta (g_j(r) - b_j)^2, g_j(r) \le 0 \\ \dfrac{\lambda_j \beta (g_j(r) - b_j)}{\lambda_j + \beta (g_j(r) - b_j)}, g_j(r) \le 0 \end{array} \right\} \tag{1.1.2}$$

The constringing speed of the generalized Lagrange Function is quicker than that of progressive projection.

2.2. Redundance Distribution Pattern

Kuo [3] solved the problem of the reliability redundance distribution by combining Lagrange multiplier with filiation delimitation.

$$\max \; f(r) = R(x_1, x_2, \cdots x_s) \tag{1.2.1}$$

Constraint condition:

$$g_j(r) \le b_j$$
$$0 < r_i \le 1$$

Here: x_i = redundance rating of component software i, $x_i \in \{1, 2, \cdots\}$

b_j = constraint limit, j = $1,2,\cdots m$

s = total of subsystems

Fyffe [4] has solved the problem of redundance distribution with dynamic programming. As for No ith subsystem, suppose there are A_i of different choices of component software, the related redundance rating will be X_i. The constraint condition of the problem is the cost and weight of the whole system, the linear combination of the cost and weight of the relative component software. The formula is as follows:

$$\max\ f(x,a)=\sum_{i=1}^{s}R_i(x_i,a_i) \quad (1.2.2)$$

Constraint condition:

$$\sum_{i=1}^{s}g_i(x_i,a_i)\le C$$

$$\sum_{i=1}^{s}w_i(x_i,a_i)\le W$$

Here: $x_i\in\{1,2,3,\cdots\}$

$a_i\in\{1,2,3,\cdots,A_i\}$

w_j = Unit Weight of component software j

C = Total Cost

W = Total Weight

2.3. Computation of Confidence Interval

The fierce social competition makes the software developers improve the efficiency constantly, with the development cycle shortening and tests decreasing. In general, very few fail data available to the component software reliability estimation, and the estimation is far from enough for the component software reliability.

Surely, there exist estimation variance and confidence interval in the result of the component software reliability, which is of great importance to the system developers. They would choose a lower estimated value of the reliability, with the estimation of the component software system being accurate (namely, the lower limit should be the result of the approaching point), rather than a higher estimated value of the reliability, with the estimation of the component software doubtful (namely, estimation variance is very great).

Buehler has made many achievements in the research of the lower limit of the reliability estimation. The software is usually equipped with two parallel independent component software, the number of the component software distributed to the test is n_1 and n_2, and the number of the tested errors is k_1 and k_2. Non-reliability of component software q_i and upper limit of the non-reliability of the system Cn_1n_2 (k_1, k_2; α) meet the following requirements [5]:

$\Pr\{0\le q_1q_2\le Cn_1n_2(k_1,k_2;\alpha)\}\ge\alpha$ (1.3.1)

Lloyd and Lipow have extended Buehler's pattern, the new pattern they put forward can be used to compute the upper limit of the reliability of the software system with any number of component software. To compute the lower limit of the confidence of the software system, which has a statistically independent component software, the method by Lindstrom and Madden is more suitable.

2.4. Shortcomings and Inadequacies

Although a great many different methods can be used to compute the reliability estimation variance and confidence interval of the system, still many methods cannot be applied to the practical problems, for these methods are with the complicated computing processes and many constraint conditions. Generally, one of the assumption should be possible.

(1) The fail time of the component software is subject to the index distribution;

(2) The fail state of the component software is subject to the binomial distribution.

It is not difficult to realized the index distribution, the index distribution can be calculated by distribution of χ^2, but not all the fail time of the component software is subject to the index distribution. Besides, the given design structure of the system is limited in many research and documentation.

It is our hope to make use of the known the reliability estimation variance of the component software rationally and scientifically, and with these data to compute the expected value and estimation variance of the system reliability.

3. DECOMPOSING SOLUTION OF SYSTEM VARIANCE

The solution of the decomposition of the system variance can be used effectively to improve the reliability estimation variance and to increase the computing efficiency and capability of the reliability confidence interval. The basic idea is that, so long as the component software reliability are independently estimated, the system variance can be decomposed into the product of the subsystem, and the subsystem is a union composed of the serial and parallel component software.

3.1. Symbol Definition

$r_i(t)$: reliability of component software i at t time

$p_i(t)$: non-reliability of component software i at t time

$R(t)$: system reliability at t time

t' : performing time

f_i : fail number observed at the time of [0, t']

n_i : number of the testing component software i at t' time

$\wedge$: indicating calculation

s: number of subsystem

3.2. System Variance Decomposition

As for the fixed system pattern of the reliability estimation and estimation variance of any known component soft-

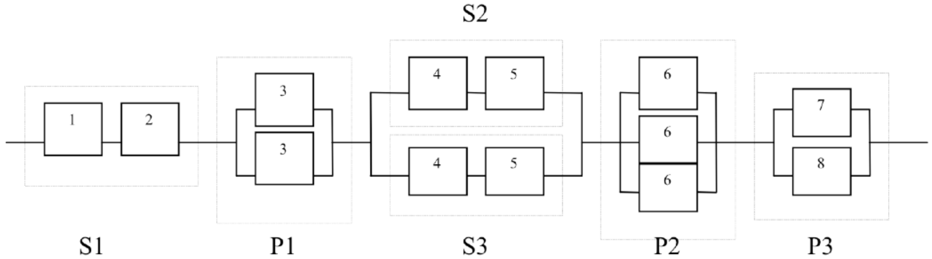

Fig. (1). System before decomposition.

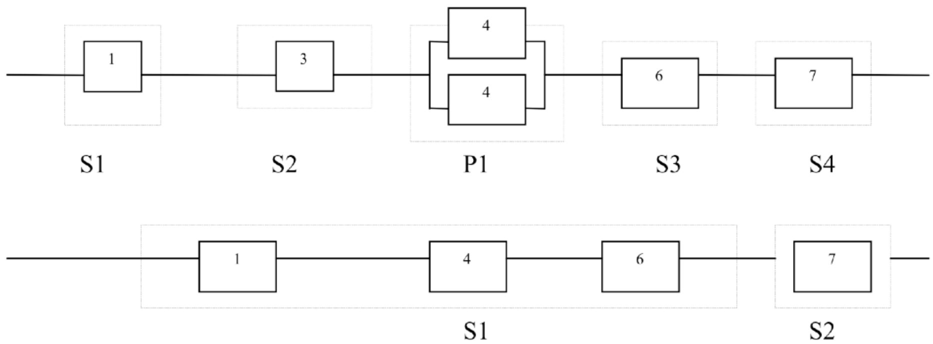

Fig. (2). System after decomposition.

ware, the estimated value of the system reliability can be computed by decomposition plan. The only restriction of the method is that the system must be broken into the elements of the serial or parallel subsystems. Most complicated systems can meet this requirement, but some systems cannot be broken into the elements of the serial or parallel subsystems elements, such as terminal network and bridging configuration, to which the variance decomposing technology cannot applied.

The following is an example of the system decomposition: Suppose a software system SP is composed of 8 component software (S_i indicates serial structure, P_i indicates parallel structure). Fig. (**1**). shows the system before decomposition, Fig. (**2**). shows the system after decomposition.

This system can be broken into 6 serial subsystems, some of the component software are used twice (for examples, 3 and 6), but these component software are statistically independent. When the component software are of various types, we can classify the available data to obtain the estimated value of the reliability of the statistically independent component software.

3.3. Compotation of the Reliability Estimation Variance of the Serial and Parallel System

As the function distribution of the component software is unknown, the random variable X_k can be used to indicate the estimated value of the component software reliability, it is statistically independent. Usually, the serial system and parallel system are indicated as follows:

Serial system: $Var\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}$ (2.2.1)

Parallel system: $Var\{1-(1-X_1)\cdot(1-X_2)\cdot\cdots\cdot(1-X_n)\}$ (2.2.2)

In actual computation, we need use the relative knowledge of probability statistics to deduce the real practical formula.

$$Var\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\} \tag{2.2.3-1}$$

$$= E\{(X_1 \cdot X_2 \cdot\cdots\cdot X_n) - E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}\}^2$$

$$= E\left\{\begin{matrix}(X_1 \cdot X_2 \cdot\cdots\cdot X_n)^2 - 2E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\} \\ \cdot(X_1 \cdot X_2 \cdot\cdots\cdot X_n) + E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}^2\end{matrix}\right\}$$

$$= \begin{matrix}E\{(X_1 \cdot X_2 \cdot\cdots\cdot X_n)^2\} - 2E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\} \\ \cdot E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\} + E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}^2\end{matrix}$$

$$= E\{(X_1 \cdot X_2 \cdot\cdots\cdot X_n)^2\} - E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}^2$$

$$E\{(X_1 \cdot X_2 \cdot\cdots\cdot X_n)^2\} - E\{X_1 \cdot X_2 \cdot\cdots\cdot X_n\}^2 \tag{2.2.3-2}$$

$$= \begin{matrix}E\{X_1^2 \cdot X_2^2 \cdot\cdots\cdot X_n^2\} \\ -[E\{X_1\}\cdot E\{X_2\}\cdot\cdots\cdot E\{X_n\}]^2\end{matrix}$$

$$= \begin{matrix}E\{X_1^2\}\cdot E\{X_2^2\}\cdot\cdots \\ \cdot E\{X_n^2\} - E\{X_1\}^2 \cdot E\{X_2\}^2 \cdot\cdots\cdot E\{X_n\}^2\end{matrix}$$

$$= \prod_{k=1}^{n} E\{X_k^2\} - \prod_{k=1}^{n} E\{X_k\}^2$$

By substituting the equation 2.2.3-2 into the equation 2.2.3-1, we can deduce the formula of the reliability estimation variance of the serial system indicated by the expected value of the reliability estimation variance of the component software.

$$Var\left\{X_1 \cdot X_2 \cdot \cdots \cdot X_n\right\} \quad (2.2.3\text{-}3)$$

$$= \prod_{k=1}^{n} E\left\{X_k^2\right\} - \prod_{k=1}^{n} E\left\{X_k\right\}^2$$

$$= \prod_{k=1}^{n}\left[Var\left\{X_k\right\} + E\left\{X_k\right\}^2\right] - \prod_{k=1}^{n} E\left\{X_k\right\}^2$$

Make $Y_k \equiv 1 - X_k$, then $E\left(Y_k\right) = E\left(1 - X_k\right) = 1 - E\left(X_k\right)$

$$Var\left(Y_k\right) = Var\left(Y_k\right) \quad (2.2.4\text{-}1)$$

$$Var\left\{1 - \left[(1-X_1)\cdot(1-X_2)\cdot\cdots\cdot(1-X_n)\right]\right\}$$

$$= Var\left\{(1-X_1)\cdot(1-X_2)\cdot\cdots\cdot(1-X_n)\right\}$$

$$= Var\left\{Y_1 \cdot Y_2 \cdot \cdots \cdot Y_n\right\}$$

$$= \prod_{k=1}^{n} E\left\{Y_k^2\right\} - \prod_{k=1}^{n} E\left\{Y_k\right\}^2$$

$$= \prod_{k=1}^{n}\left[Var\left\{Y_k\right\} + E\left\{Y_k\right\}^2\right] - \prod_{k=1}^{n} E\left\{Y_k\right\}^2 \quad (2.2.4\text{-}2)$$

By substituting the equation2.2.4-2 into the equation 2.2.4-1 we can deduce the formula of the reliability estimation variance of the parallel system indicated by the expected value of the reliability estimation variance of the component software.

$$Var\left\{1 - \left[(1-X_1)\cdot(1-X_2)\cdot\cdots\cdot(1-X_n)\right]\right\}$$

$$= \prod_{k=1}^{n}\left[Var\left\{X_k\right\} + \left(1 - E\left\{X_k\right\}\right)^2\right] - \prod_{k=1}^{n}\left(1 - E\left\{X_k\right\}\right)^2 \quad (2.2.4\text{-}3)$$

Finally, we can make an assumption as follows:

If $X_k = \hat{r}_i(t')$, then $\mu_k = r_i(t')$, $\sigma_i^2 = Var(\hat{r}_i(t'))$

So, we can deduce the general formula of the reliability estimation variance of the system.

$$Var\left\{X_1 \cdot X_2 \cdot \cdots \cdot X_n\right\}$$

$$= \prod_{k=1}^{n}(\sigma_k^2 + \mu_k^2) - \prod_{k=1}^{n}\mu_k^2 \quad (2.2.5)$$

$$Var\left\{1 - \left[(1-X_1)\cdot(1-X_2)\cdot\cdots\cdot(1-X_n)\right]\right\}$$

$$= \prod_{k=1}^{n}[\sigma_k^2 + (1-\mu_i)^2] - \prod_{k=1}^{n}(1-\mu_i)^2 \quad (2.2.6)$$

4. COMPUTATION OF RELIABILITY ESTIMATION VARIANCE OF COMPONENT SOFTWARE

4.1. Computation of Estimation Variance of Component Software

In computation of the reliability estimation variance of the system, it is the the first step to compute the unbiased estimated value and estimation variance. As there is no definition for the fail time distribution and component distribution, we can only make use of the available data to estimate the reliability of the component software. Hence $r_i(t')$ is an unknown constant, the estimated value of the component software reliability is a variable, it can only be roughly defined with the estimated value and variance of $r_i(t')$

In accordance with the symbol definition in 2.1, it is known that, as to the component software i, n_i is the number of the component software i at the tested time t', f_i is the fail number of the component software i tested at the time of [0, t']. If the (successful / failing) state of the component software is subject to the Bernoulli test, with statistically independent parameter $r_i(t')$, the unbiased estimation and approximate estimation variance can be defined by the following binomial distribution [6].

$$\hat{p}_i(t') = \frac{f_i}{n_i}$$

$$\hat{r}_i(t') = 1 - \hat{p}_i(t') = \frac{n_i - f_i}{n_i}$$

$$Var\left\{\hat{r}_i(t')\right\} = \frac{r_i(t') \cdot p_i(t')}{n_i} \quad V\hat{a}r\left\{\hat{r}_i(t')\right\} = \frac{\hat{r}_i(t') \cdot \hat{p}_i(t')}{n_i}$$

It is of great importance that the fail time of the component software is not defined here. At the early stage of designing, it is necessary to estimate the reliability of the expected system, but at this stage, there are few data available, it is difficult to choose a proper fail time distribution of the component software, and an erroneous assumption of the distribution will directly affect the final design.

4.2. Computation of Reliability Estimation Variance of Software

Obtaining the estimated value and estimation variance of the component software reliability, it is easy to compute the reliability estimation variance of the software system, with the help of the above-mentioned formula of the reliability estimation variance of the software system (with R_S indicating the reliability of the serial software system; R_P indicating the reliability of the parallel software system)

$$Var\left\{\hat{R}_S(t')\right\} = \prod_i\left[Var\left\{\hat{r}_i(t')\right\} + r_i(t')^2\right] - \prod_i r_i(t')^2 \quad (3.2.1)$$

$$Var\left\{\hat{R}_P(t')\right\} = \prod_i\left[Var\left\{\hat{r}_i(t')\right\} + p_i(t')^2\right] - \prod_i p_i(t')^2 \quad (3.2.2)$$

The main steps of the reliability variance decomposition process of the complicated software system are as follows:

(1) To define the pattern of the software reliability (diagram) and the reliability of all the component software used in the system.

(2) To compute the estimated value and estimation variance of the component software reliability with the available data.

(3) To break the software into parts, each part composed of component software, in the serial or parallel structure, with the serial marked as S_i, and the parallel marked as P_i.

(4) To compute the estimated value and estimation variance of the reliability of each serial and parallel part, with the equations of 3.2.1 and 3.2.2.

(5) To substitute each part with an equivalent assumed component software.

(6) To repeat steps 3 - 5 till the reliability pattern of the software is substituted by a single component software, the reliability estimation variance of the original software system can be roughly defined with the reliability estimation variance of the component software.

CONCLUSION

With the wide application of the component software in the software development, the increasing importance and in-depth study has been attached to it. The estimation indexes of the component software quality is same as that of the ordinary software, it should meet the requirement of the qualitative indexes, such as, portability, maintainability, efficiency, manageability, reliability, and functionality. In these qualitative indexes, the reliability remains the most important. So an ever increasing attention has been drawn on the relative study of the component software reliability. And the system variance decomposition is an easy and practicable computation, it shows a prospect of a wide application, by avoiding the complicate mathematical probability computation and the required condition decomposing the reliability estimation variance of the software system.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] D. Hamlet, D. Mason, and D. W. Coit. "Theory of software reliability based on components," In: *Proceedings of the 3rd International Workshop on Component-Based Software Engineering, IEEE, Computer Society*, Toronto, pp. 361-370, 2001.

[2] C. L. Hwang, F. A. Tillman, and W. Kuo, "Reliability optimization by generalized lagrangian-function and reduced-gradient methods," *IEEE Transactions on Reliability*, vol. 28, no. 4, pp. 79-87, 1979.

[3] W. Kuo, H. Lin, Z. Xu, and W. Zhang, "Reliability optimization with the lagrange multiplier and branch and bound technique," *IEEE Transactions on Reliability*, vol. 36, no. 5, pp. 624-630, 1979.

[4] D. E. Fyffe, W.W. Hines, and N.K. Lee, "System reliability allocation problem and a computational algorithm," *IEEE Transactions on Reliability*, vol. 17, no. 5, pp. 64-69, 1968.

[5] T. Jin, "*Complex System Reliability Analysis and Optimizati-on Considering Component Reliability Estimation Uncertainty*," Ph.D Thesis, UMI-3043040, University of New Brunswick, New Jersey, 2001.

[6] T. Jin, and D. W. Coit, "Variance of system-reliability estimates with arbitrarily repeated components," *IEEE Tranactions on Reliability*, vol. 50, no. 4, pp. 409-413, 2001.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014