Math. Struct. in Comp. Science (2014), vol. 24, iss. 5, e240508, 22 pages.
doi:10.1017/S0960129513000789

Bayesian theory based software reliability demonstration test method for safety critical software†

YUMEI WU, RISHENG YANG, HAIFENG LI and MINYA LU

School of Reliability and Systems Engineering,
Beihang University, Beijing, China
Email: wuyumei@buaa.edu.cn

Received 20 July 2011; revised 18 August 2013

The original software reliability demonstration test (SRDT) does not take adequate account of prior knowledge or the prior distribution, which can lead to an expensive use of many resources. In the current paper, we propose a new improved Bayesian based SRDT method. We begin by constructing a framework for the SRDT scheme, then we use decreasing functions to construct the prior distribution density functions for both discrete and continuous safety-critical software, and then present schemes for both discrete and continuous Bayesian software demonstration functions (which we call DBSDF and CBSDF, respectively). We have carried out a set of experiments comparing our new schemes with the classic demonstration testing scheme on several published data sets. The results reveal that the DBSDF and CBSDF schemes are both more efficient and more applicable, and this is especially the case for safety-critical software with high reliability requirements.

1. Introduction

The failure or incorrect running of safety critical software can result in catastrophic loss of life and property. Therefore, the reliability of safety critical software has become the most important feature of software quality (Lyu 1996). Software reliability demonstration testing (SRDT) is applied to validate the reliability level of the software under test, and the related reliability metrics (such as the failure probability or failure rate) can be acquired by SRDT. There are currently several statistically based SRDT schemes for calculating the required testing duration or testing cases, such as Laplace's succession rule (Tal *et al.* 2000), TRW software reliability theory (Thayer *et al.* 1978), Bayesian methods with or without prior information (Littlewood and David 1997; Qin *et al.* 2005; Yang *et al.* 2004; Qin *et al.* 2008; Dey and Rao 2005; Lindley and Smith 1972; Lyu 1996; Rahrouh 2005; Qin and Lei 2004; Miller *et al.* 1992), probability ratio sequential testing (Department of Defense 1996) and single risk sequential testing (Tal *et al.* 2001). Many schemes use fixed-duration SRDT testing, such as TRW and Laplace's rule, and are not

† This work was partially supported by Project Z231020 of the Ministry of Industry and Information Technology of China.

suitable for safety critical software because the required testing effort for these schemes is very large.

A Bayesian theory based software reliability demonstration test (BSRDT) method was first proposed in Littlewood and David (1997). Since then, it has come to be regarded as the most popular and effective method for reliability demonstration testing of safety critical software, and has been successfully applied to some safety critical software in the astronautical field. Some research has shown that if valid prior information can be acquired, the estimation of software reliability metrics can be calculated accurately and the number of testing cases or the testing duration time can be decreased. This is of great significance in improving the efficiency of SRDT for safety critical software as well as in enhancing and promoting its use in that area.

It can be seen that the construction of the prior distribution function for software reliability metrics has a critical role in determining the details and validity of BSRDT schemes (Qin *et al.* 2005; Han 2004; Littlewood and David 1997). Existing BSRDT schemes determine the prior distribution function for software reliability metrics using a conjugate distribution method, which is designed to simplify the derivation process of the corresponding posterior distribution function (Thayer *et al.* 1978). However, the conjugate distribution method does not take account of the fact that safety critical software has high reliability requirements (for example, the failure rate must be less than or equal to 10^{-3} or 10^{-5}). As a result, the prior distribution function provided by the conjugate distribution method may not be suitable for describing the prior distribution of the reliability metrics for safety critical software.

The decreasing function method is a new method for constructing the prior distribution function. The core idea of this is to select a decreasing function of the reliability metrics as their prior distribution function, which accords with the fact that for safety critical software, the probability of the reliability metrics being large is small and the probability of the reliability metrics being small is large. This suggests that the decreasing function method is more suitable for constructing the prior distribution function for the reliability metrics of high reliability safety critical software.

In the current paper, we first use the decreasing function method to construct the prior distribution function for the reliability metrics based on an existing BSRDT scheme, and then propose BSRDT schemes with decreasing functions for both discrete and continuous Bayesian software demonstration functions (DBSDF and CBSDF, respectively).

2. Framework for constructing BSRDT schemes

The basic idea of a BSRDT scheme is to determine the prior distribution function of the software reliability metrics that need be demonstrated and used to determine the posterior distribution function. The remaining details of the BSRDT scheme can then be given. With this in mind, our framework for constructing BSRDT scheme is as follows.

First we assume that the cumulative detected failure number

$$X \; F(x;\theta), \quad \theta \in \Theta,$$

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

where $f(x,\theta)$ is the probability density function and θ denotes the reliability metrics to be demonstrated (for example, θ for continuous software could be chosen to be the failure rate, or MTTF, and θ for discrete software could be chosen to be the failure probability). We assume that the prior distribution function of θ is $H(\theta)$ and the corresponding density function is $h(\theta)$. The joint probability density function of (X,θ) is then

$$f(x,\theta) = f(x|\theta) \cdot h(\theta). \tag{1}$$

The marginal distribution of the cumulative detected failure number X is then

$$g(x) = \int_{\Theta} f(x|\theta) \cdot h(\theta) d\theta. \tag{2}$$

According to Bayesian theory, given X, the conditional probability density function of θ is

$$\begin{aligned} h(\theta|x) &= \frac{f(x,\theta)}{g(x)} \\ &= \frac{f(x,\theta) \cdot h(\theta)}{\int f(x,\theta) \cdot h(\theta) d\theta}, \end{aligned} \tag{3}$$

where $h(\theta|x)$ denotes the posterior probability density function of θ.

If there is no prior information for the demonstration test scheme, the prior probability density function $h(\theta)$ can be determined on the basis of expert experience. However, if we do have some prior information for a demonstration test scheme, the parameters of $h(\theta)$ can be estimated by statistical methods based on historical failure data, and then the prior probability density function $h(\theta)$ can be obtained.

Given the software reliability demonstration index (θ_0, c, r), where θ_0 is the maximum acceptable failure rate or failure probability, c represents the confidence level and r is the tolerance number of failures during SRDT, we can obtain concrete forms for the discrete and continuous BSRDT as follows:

— For continuous software, the required time T for SRDT is the smallest t satisfying

$$\begin{aligned} (\theta \leqslant \theta_0) &= \int_0^{\theta_0} h(\theta|r,t) d\theta \\ &\geqslant c. \end{aligned} \tag{4}$$

— For discrete software, the required number of test cases (that is, N) for SRDT is the smallest n satisfying

$$\begin{aligned} P(\theta \leqslant \theta_0) &= \int_0^{\theta_0} h(\theta|r,n) d\theta \\ &\geqslant c. \end{aligned} \tag{5}$$

3. BSRDT scheme with a decreasing function

3.1. *Selecting a prior distribution function based on a decreasing function*

We will begin by introducing the concept of the kernel of a distribution density function (Dey and Rao 2005).

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Definition 3.1. Assume that $f(x)$ is the probability density function of a random variable X. Then, if

$$f(x) = cg(x),$$

where c is a constant independent of x and $g(x)$ is the part dependent on x, we say $g(x)$ is the kernel of $f(x)$, and we write

$$f(x) \propto g(x).$$

Thus, selecting a prior distribution function based on a decreasing function means selecting a decreasing function of the reliability metrics θ (that is, the failure rate or failure probability) as the kernel of the prior probability density function of θ.

3.1.1. *Prior distribution function of reliability metrics for continuous software.* Assuming that the failure rate λ of some continuous software is a random variable, the kernel of the prior probability density function of λ can be chosen as a representative decreasing function of λ in the form $e^{-a\lambda}$, that is, $h(\lambda) \propto e^{-a\lambda}$ (here, a is the hyper-parameter to be estimated). As a result, the prior probability density function of λ can be given as

$$f(\lambda) = Ae^{-a\lambda}. \tag{6}$$

For probability density functions, we have

$$\begin{aligned}\int_0^{+\infty} f(\lambda)d\lambda &= \int_0^{+\infty} Ae^{-a\lambda}d\lambda \\ &= 1,\end{aligned}$$

so $A = a$.

We now assume that the probability of the failure number x during time interval $(0, t]$ is equal to k, which is the conditional probability of failure rate, and that it follows a Poisson distribution with parameter λt (Littlewood and David 1997). We thus obtain

$$P(x = k|\lambda) = (\lambda t)^k e^{-\lambda t}/k!.$$

Combining this with Equation (6), the joint probability density function of failure number x and failure rate λ is

$$g(x = k, \lambda) = a\frac{(\lambda t)^k}{k!}e^{-\lambda(a+t)}. \tag{7}$$

According to Equation (7), the marginal probability density function of failure number x is then

$$\begin{aligned}g(x = k) &= \int_0^{+\infty} g(x = k, \lambda)d\lambda \\ &= \int_0^{+\infty} a\frac{(\lambda t)^k}{k!}e^{-\lambda(a+t)}d\lambda \\ &= \frac{at^k}{k!}\int_0^{+\infty} \lambda^k e^{-(a+t)\lambda}d\lambda \\ &= \frac{at^k}{k!}\frac{\Gamma(k+1)}{(a+t)^{k+1}}\end{aligned} \tag{8}$$

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

where $\Gamma(x)$ is the Gamma function:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt. \tag{9}$$

If x is a positive integer, then $\Gamma(x) = (x-1)!$, so Equation (8) can be simplified to

$$g(x = k) = \frac{at^k}{(a+t)^{k+1}}. \tag{10}$$

If r failures are observed after the software has been running continuously for a period of time t, then the posterior probability density function of failure rate λ is given by

$$\begin{aligned} f(\lambda|r,t,a) &= \frac{g(x=r,\lambda)}{g(x=r)} \\ &= \frac{(a+t)^{r+1}}{r!}\lambda^r e^{-\lambda(a+t)}. \end{aligned} \tag{11}$$

3.1.2. *Prior distribution function of reliability metrics for discrete software.* As with the continuous software case, we use the decreasing function method to define the kernel of the prior probability density function for discrete software reliability metrics. Specifically, the failure probability p can be assigned a representative decreasing function of p in the form $(1-p)^a$. As a result, the prior probability density function of p is given by

$$f(p) = A(1-p)^a. \tag{12}$$

Moreover, for probability density functions, we have

$$\int f(p)dp = \int A(1-p)^a dp,$$

so we obtain $A = a + 1$, and Equation (12) can be rewritten as

$$f(p) = (a+1)(1-p)^a. \tag{13}$$

We now assume that successive runs of the discrete software are statistically independent Bernoulli trials. Let p be the failure probability of one run randomly selected from the operation profile. Thus, given p, the number of failures r in n runs obeys the Binomial distribution (Littlewood and David 1997), that is,

$$P(r,n|p) = C_n^r p^r (1-p)^{n-r}.$$

Combining this with Equation (13), the joint probability density function of failure number r and failure probability p is then given by

$$\begin{aligned} g(r,n,p) &= P(r,n|p) \cdot f(p) \\ &= (a+1)C_n^r p^r p^r (1-p)^{a+n-r}. \end{aligned} \tag{14}$$

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

The marginal probability density function of the failure probability p is then

$$\begin{aligned} g(p) &= \int_0^1 g(r,n,p)dp \\ &= (a+1)C_n^r \int_0^1 p^r(1-p)^{a+n-r} \\ &= (a+1)C_n^r B(r+1, a+1+n-r). \end{aligned} \tag{15}$$

If r failures are observed after n test cases have been executed, then the posterior probability density function of failure probability n is given by

$$\begin{aligned} f(p|r,n,p) &= \frac{g(r,n,p)}{g(r,n)} \\ &= \frac{p^r(1-p)^{a+n-r}}{B(r+1, a+1+n-r)} \\ &= \text{Beta}\,(r+1, a+1+n-r). \end{aligned} \tag{16}$$

3.2. *A continuous BSRDT scheme with a decreasing function (CBSDF)*

Safety critical software has usually been subjected to a long period of software reliability growth testing before it is submitted to an SRDT. The time between failures data (denoted by T_1, L, T_n) collected during software reliability growth testing form the most dependable and useful prior information. Therefore, for the current paper, we chose to use the time between failure data collected from the later stages of reliability growth testing as the prior information to maximise the accuracy of the estimates of the value of the hyper-parameter a in the prior probability density function of failure rate λ. We will now give the details of this approach to determining the hyper-parameter a in Equation (6) using failure information.

The mathematical expectation of the failure number X of the software during time interval $(0, 1]$ according to Equation (10) is

$$\begin{aligned} E(X) &= \sum_{r=0}^{+\infty} r \cdot g(x=r) \\ &= \sum_{r=0}^{+\infty} \frac{art^r}{(a+t)^r} \\ &= \frac{t}{a}. \end{aligned} \tag{17}$$

From Equation (17), we can see that the value of the hyper-parameter a can be determined by time t and $E(X)$ (that is, the expectation of the failure number during the time interval $(0, 1]$). Hence, we need to convert the sequence of samples of the time between failures $(T_1, T_2, \ldots, T_n)$ to a sequence of samples of failure numbers so that the estimated value of the hyper-parameter a can be obtained from Equation (17).

We now assume that t_φ is a comparative large time value (t_φ should be larger than the sequence of time between failures $(T_1, T_2, \ldots, T_n)$). The corresponding failure number

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

during the time interval $(0, t_\varphi]$ (denoted by s_i) is then t_φ / T_i. So we can convert $T_1, T_2, \ldots, T_n$ to a sequence of failure numbers as follows:

$$\{s_i\}_{i=1}^n = (t_\varphi / T_i)_{i=1}^n. \tag{18}$$

From Equations (17) and (18), we get

$$\begin{aligned} E(x) &= \frac{1}{n}\sum_{i=1}^{n} s_i \\ &= t/a. \end{aligned}$$

The estimated value of the hyper-parameter a is then

$$a = \frac{t_\varphi}{\frac{1}{n}\sum_{i=1}^{n} s_i}. \tag{19}$$

Using Equation (18), this can be simplified to

$$a = \frac{n}{\sum_{i=1}^{n} \frac{1}{T_i}}. \tag{20}$$

From Equation (20), we can see that the value of t_φ will not influence the estimated value of the hyper-parameter a. From the above, the prior probability density function of failure rate λ can be given by

$$f(\lambda) = ae^{-a\lambda}.$$

And from Equation (11), the posterior probability density function of λ is then

$$f(\lambda \mid r, t, a) = \frac{(a+t)^{r+1}}{r!}\lambda^r e^{-\lambda(a+t)}. \tag{21}$$

According to Equation (21), given the SRDT index (λ_0, c, r), the required time T for SRDT is the smallest t satisfying

$$\begin{aligned} P(\lambda \leqslant \lambda_0) &= \int_0^{\lambda_0} f(\lambda \mid r, t, a)d\lambda \\ &= \int_0^{\lambda_0} \frac{(\hat{a}+t)^{r+1}}{r!}\lambda^r e^{-\lambda(\hat{a}+t)}d\lambda \\ &\geqslant c. \end{aligned} \tag{22}$$

In particular, if we set the acceptable number of failures to 0, the time required for SRDT can be obtained by solving the equation

$$\begin{aligned} P(\lambda \leqslant \lambda_0) &= \int_0^{\lambda_0} f(\lambda \mid 0, t, a)d\lambda \\ &= \int_0^{\lambda_0} (\hat{a}+t)^{r+1}\lambda^r e^{-\lambda(\hat{a}+t)}d\lambda \\ &\geqslant c. \end{aligned} \tag{23}$$

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

3.3. *A discrete BSRDT scheme with a decreasing function (DBSDF)*

After n test cases have been executed, the expectation of the number of observed failures x according to Equation (15) is

$$\begin{aligned} E(X) &= \sum_{r=0}^{n} r \cdot g(x=r) \\ &= \sum_{r=0}^{n} r \cdot \int_0^1 (a+1) C_n^r p^r p^r (1-p)^{a+n-r} dp \\ &= (a+1) \int_0^1 (1-p)^a \left\{ \sum_{r=0}^{n} r C_n^r p^r p^r (1-p)^{n-r} \right\} dp \\ &= (a+1) \int_0^1 np(1-p)^a dp \\ &= n(a+1)B(2, a+1) \\ &= \frac{n}{a+2}. \end{aligned} \tag{24}$$

We now select m groups of operation records for test cases collected in the later stages of the reliability growth testing as the prior information. If each group contains d test cases and the number of test cases that result in failures in each group is denoted by $s_1, s_2, \ldots, s_m$, we set

$$\begin{aligned} n &= d \\ s &= \frac{\sum_{i=1}^{m} s_i}{m}. \end{aligned} \tag{25}$$

From Equations (25) and (24), combined with $s = E(X)$, the estimated value of the hyper-parameter a can now be given as

$$a = \frac{d}{\sum_{i=1}^{m} s_i / m} - 2. \tag{26}$$

With the estimated value of the hyper-parameter of failure probability $p's$ prior probability density function having been obtained as $a = a$, the prior probability density function for the failure probability $p's$ is

$$f(p) = (a+1)(1-p)^a. \tag{27}$$

The posterior probability density function of p is then

$$f(p \mid r, r) = \text{Beta}\,(r+1, a+1+n-r). \tag{28}$$

Thus, if the required level of p is p_0 with confidence level c, and the tolerance number of failures during SRDT is r, the required number (that is, N) of test cases for SRDT is the

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

smallest n satisfying

$$\begin{aligned} P(p \leqslant p_0) &= \int_0^{p_0} f(p \mid r, n)dp \\ &= \int_0^{p_0} \frac{p^r(1-p)^{a+n-r}}{B(r+1, a+1+n-r)}dp \\ &\geqslant c. \end{aligned} \tag{29}$$

In particular, when $r = 0$, the required number of test cases for SRDT can be obtained by solving the equation

$$\begin{aligned} P(p \leqslant p_0) &= \int_0^{p_0} f(p \mid 0, n)dp \\ &= \int_0^{p_0} \frac{p^r(1-p)^{a+n}}{B(1, a+1+n)}dp \\ &\geqslant c. \end{aligned} \tag{30}$$

4. Case studies

In order to carry out some case studies, we selected two failure data sets collected from a safety critical software reliability growth test as the prior information for use in determining the prior probability density function. To verify the feasibility and effectiveness of the proposed CBSDF and DBSDF schemes, we applied them to the two selected failure data sets and compared the results with two typical continuous and discrete BSRDT schemes:

— The two typical continuous BSRDT schemes are:

- the continuous BSRDT scheme without prior information proposed in Lindley and Smith (1972), Qin *et al.* (2005), Yang *et al.* (2004), Qin *et al.* (2008), Dey and Rao (2005), Lyu (1996) and Rahrouh (2005), and which we will refer to as CBS1;
- the continuous BSRDT scheme with prior information proposed in Yang *et al.* (2004), Qin *et al.* (2008), Dey and Rao (2005), Lindley and Smith (1972), Lyu (1996), Rahrouh (2005) and Qin and Lei (2004), and which we will refer to as CBS2.

— The two typical discrete BSRDT schemes are:

- the discrete BSRDT scheme without prior information proposed in Cukic and Chakravarthy (2000), Qin *et al.* (2005), Han (2004), Littlewood and David (1997), Qin *et al.* (2005), Yang *et al.* (2004), Qin *et al.* (2008), Dey and Rao (2005), Lindley and Smith (1972), Lyu (1996), Rahrouh (2005), Qin and Lei (2004) and Miller *et al.* (1992), and which we will refer to as DBS1;
- the discrete BSRDT scheme with prior information proposed in Qin *et al.* (2005), Han (2004) and Littlewood and David (1997), and which we will refer to as DBS2.

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 1. *The two data sets giving the prior information (the unit of time is hours)*

SYS2		SYS2		Ckyjm		Ckyjm	
T_i	s_i	T_i	s_i	T_i	s_i	T_i	s_i
2175	45	490	204	288	347	660	151
1866	53	1194	83	1169	85	209	478
2716	36	994	100	1061	94	361	277
1520	65	3281	30	142	704	688	145
725	137	3902	25	494	202	1046	95

4.1. *Validation for CBSDF*

4.1.1. *Sources of the prior information.* The sources of the prior information are listed in Table 1. These were selected from two real failure data sets 'SYS2' and 'Ckyjm' (Lyu 1996), and represent the data for the last ten times between failures $T_1, T_2, \ldots, T_{10}$.

These sets were collected from the reliability growth tests for some real-time control system software, which was considered suitable for this case study. We assume that t_φ is 100,000 hours, which is larger than $T_1, T_2, \ldots, T_{10}$ for these two data sets. The sequence of empirical failure numbers corresponding to $T_1, T_2, \ldots, T_{10}$ is then

$$\{s_i\}_{i=1}^{1}0 = \{t_\varphi / T_i\}_{i=1}^{1}0,$$

as listed in Table 1.

4.1.2. *SRDT index.* Bearing in mind that a feature of safety critical software is that the failure rate λ_0 is between 10^{-5} and 10^{-3}, and the properties of the data sets (for example, the range of values of the estimates of the failure rate for the data sets), the index of the failure rate λ_0 was set to 10^{-3} for SYS2 and 10^{-4} for Ckyjm. The confidence level c was set to 0.99, and the tolerance number for failures r was set variously to 0, 1, 2, 3, 4 and 5.

4.1.3. *Results and results analysis.* In this section we discuss the results of the case study for the SYS2 and Ckyjm datasets:

— **Results and results analysis for SYS2:**
Given the prior information of the failure data set SYS2 listed in Table 1, the estimated values of the hyper-parameters of the prior probability density function for the CBS2 scheme are $a = 2.2$ and $b = 2793$, so the prior probability density function for CBS2 is

$$h(\lambda) = \frac{2793^{2.2}\lambda^{1.2}e^{-2793\lambda}}{\Gamma(2.2)}. \tag{31}$$

In the same way, the estimated value of the hyper-parameter of the prior probability density function for the CBSDF scheme is $a = 1285$, so the prior probability density function for CBSDF is

$$h(\lambda) = 1285e^{-1285\lambda}. \tag{32}$$

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 2. *The required SRDT time for each scheme for the SYS2 data set*

r	CBS1	CBS2	CBSDF
0	4605.2	4164.3	3319.8
1	6638.4	5903.9	5353.0
2	8405.9	7526.9	7120.0
3	10045.1	9075.0	8759.0
4	11604.6	10571.1	10319.3
5	13108.5	12027.1	11823.1

Using the SRDT index $(0.001, 0.99, r)$ set above, we can then determine the required SRDT time for the CBS1, CBS2 and CBSDF schemes using the data set SYS2 with r set variously to 0, 1, 2, 3, 4 and 5. The results are given in Table 2.

Table 2 shows that for each value of r, the required SRDT times calculated by the proposed CBSDF scheme are all smaller than those calculated by CBS1 and CBS2. For example, when $r = 0$, the required SRDT time using CBS1 and CBS2 are 4,605.2 and 4,164.2 hours, while the required SRDT time using CBSDF is only 3,319.8 hours. This shows that the CBSDF scheme decreases the required SRDT time by 1,285.4 hours (that is, 28%) and 844.5 hours (that is, 20%) compared with CBS1 and CBS2, respectively.

— **Results and results analysis for data set Ckyjm:**
In the same way as for the previous case, given the prior information of the failure data set Ckyjm listed in Table 1, the estimated values of the hyper-parameters of the prior probability density function for the CBS2 scheme are $a = 2.9$ and $b = 483$, so the prior probability density function of CBS2 is

$$h(\lambda) = \frac{483^{2.9}\lambda^{1.9}e^{-483\lambda}}{\Gamma(2.9)}. \tag{33}$$

And the estimated value of the hyper-parameter of the prior probability density function for the CBSDF scheme is $a = 163$, so the prior probability density function for CBSDF is

$$h(\lambda) = 163e^{-163\lambda}. \tag{34}$$

Using the SRDT index $(0.001, 0.99, r)$ set above, we can then determine the required SRDT time for the CBS1, CBS2 and CBSDF schemes using the data set Ckyjm with r set variously to 0, 1, 2, 3, 4 and 5. The results are given in Table 3.

From Table 3, the required SRDT times calculated by the proposed CBSDF scheme are again all smaller than those calculated by the CBS1 and CBS2 schemes for each value of r. For example, when $r = 0$, the required SRDT time when using CBS1 and CBS2 are 46,051.7 and 82,786 hours, respectively, while the required SRDT time using CBSDF is only 45,888.3 hours. That is, the CBSDF scheme decreases the required SRDT time by 0.3% and 45%, comparedwith CBS1 and CBS2, respectively. This

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 3. *The required SRDT time for each scheme for the Ckyjm data set*

r	CBS1	CBS2	CBSDF
0	46051.7	82786.0	45888.3
1	66383.6	99223.8	66220.1
2	84059.5	114949.6	83896.1
3	100451.2	129910.5	100287.8
4	11604.6	144549.1	115882.8
5	131084.9	158856.3	130921.4

means that the proposed CBSDF is more effective than both CBS1 and CBS2 in terms of decreasing the required SRDT time.

However, the difference between CBS1 and CBSDF is trivial for this failure data set, so we suggest that if the prior information cannot be used accurately by the appropriate prior distribution of failure probability, the Bayesian-based scheme with prior information may yield a worse result compared with the scheme without prior information. Thus, selecting an appropriate prior distribution of failure probability for the failure data set is very significant in determining the effectiveness of the SRDT scheme.

4.1.4. *Analysis of the applicability of the CBSDF scheme.* To provide further evidence that CBSDF is a more suitable scheme for use with continuous high reliable safety-critical software (that is, with an index of failure rate ranging between 10^{-5} and 10^{-3}), we calculated the results for the CBS1, CBS2 and CBSDF schemes as the index of failure rate varied from 10^{-5} to 10^{-3} with a step length of 4.95×10^{-5}.

For this test, the SRDT confidence level c was set to 0.99 and the tolerance number of failures r was set to 0.

Figure 1 shows the time required for SRDT (vertical axis) using the CBS2 and CBSDF schemes for different failure rates λ_0 (horizontal axis) with prior information taken from the SYS data set. Figure 2 shows the corresponding results for the Ckyjm data set.

The performance of the CBSDF, CBS2 and CBS1 schemes with different index of failure rates was then evaluated by computing the reduction in the required SRDT time when using the CBSDF and CBS2 schemes compared with CBS1. The results are plotted in Figures 3 and 4 for the SYS2 and Ckyjm data sets, respectively, where the horizontal axis represents different failure rate λ_0, and the vertical axis indicates the reduction in the required SRDT time compared with the CBS1 scheme – the line $y = 0$ can be viewed as the baseline CBS1 scheme.

Summarising the results:

— Figures 1 and 2 show that:

(1) When the value of λ_0 is increased from 10^{-5} to $10^{-}3$, the required SRDT time for the CBS2 and CBSDF schemes shows a decreasing trend. This is consistent with

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

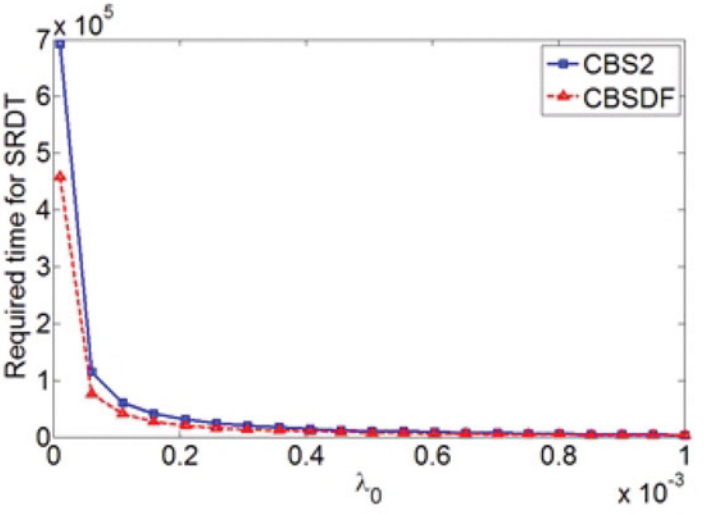

Fig. 1. (Colour online) Required SRDT time when using CBS2 and CBSDF (SYS2)

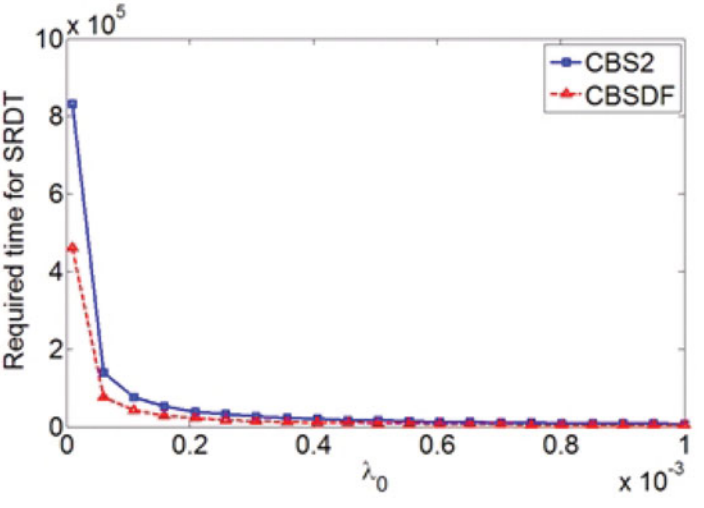

Fig. 2. (Colour online) Required SRDT time when using CBS2 and CBSDF (Ckyjm)

the obvious conclusion that the greater the index of failure rate, the shorter the required SRDT time.

(2) When the index of failure rate is very small, specifically, when

$$10^{-5} < \lambda_0 < 8 \times 10^{-5},$$

the required SRDT time when using CBSDF is much less than that for CBS2. However, as the value of λ_0 is gradually increased, the CBSDF curve tends to be moderate, so the gap between the required SRDT time for CBSDF and CBS2 decreases. This shows that CBSDF is more effective for continuous high reliability safety-critical software than CBS2.

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

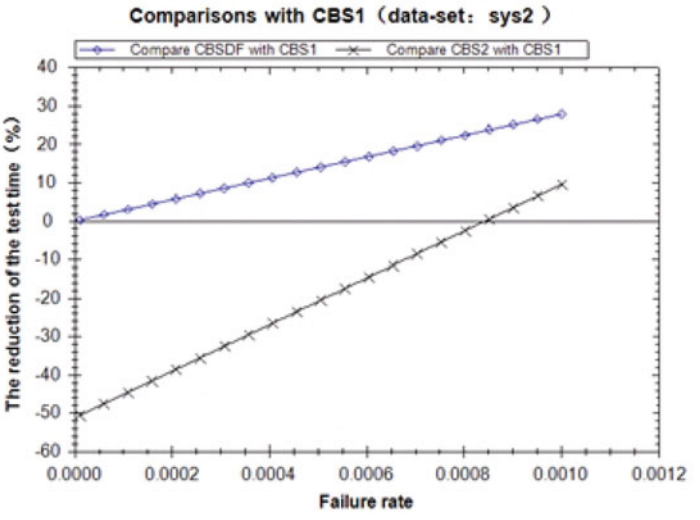

Fig. 3. (Colour online) Comparing CBSDF and CBS2 with CBS1 (SYS2)

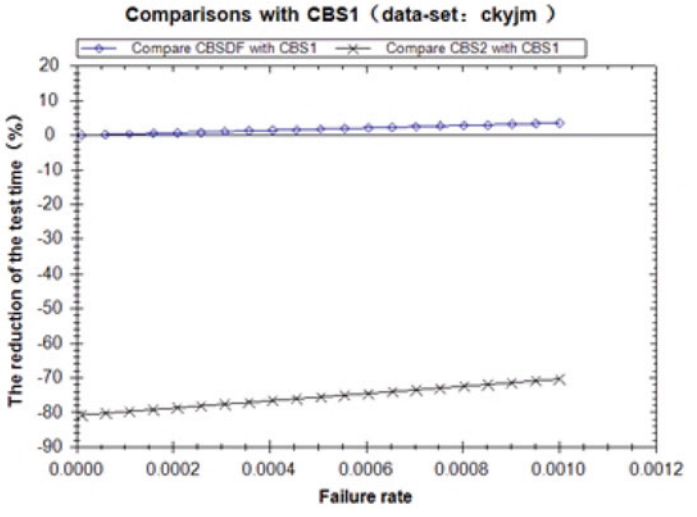

Fig. 4. (Colour online) Comparing CBSDF and CBS2 with CBS1 (Ckyjm)

— Figures 3 and 4 show that:

(1) When $\lambda_0 \in [10^{-5}, 10^{-3}]$, the curve for CBSDF is always above both the baseline of the CBS1 scheme without prior information and the CBS2 curve. In other words, the required SRDT time when using CBSDF is always smaller than that for CBS1 and CBS2. This means that CBSDF is more effective than CBS1 and CBS2 in terms of effectiveness and applicability.

(2) When $\lambda_0 \in [0.0009, 0.001]$, the schemes with prior information (that is, CBSDF and CBS2) require a markedly shorter SRDT time than the scheme without prior information (that is, CBS1) – see Figure 3. For example, when $\lambda_0 = 0.001$, the CBSDF scheme gives a 28% saving in test effort and CBS2 saves 8%.

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 4. *The prior information data sets*

CSR1		CSR1		xrzmo		xrzmo	
T_i	k_i	T_i	k_i	T_i	s_i	T_i	s_i
1236	80	586	170	3186	347	660	151
1406	71	1008	99	571	85	209	478
1471	67	3100	32	563	94	361	277
1749	57	2686	37	2770	704	688	145
2167	46	1893	52	652	202	1046	95

(3) However, as shown in Figure 4, when the reliability index is comparatively high, that is, $\lambda_0 \in [0.00001, 0.0009]$, although the CBSDF scheme is still better than both CBS1 and CBS2, the required SRDT time for CBS2 is significantly longer than for CBS1. For example, when $\lambda_0 = 0.00001$, CBS2 requires an SRDT time that is up to 80% longer than CBS1.

(4) For the same SRDT index and using the same prior information, the CBSDF scheme is obviously superior to CBS2. One potential reason for this is that the CBS2 scheme's prior probability density function is a Gamma function $\Gamma(a, b)$, which becomes a decreasing function only when $0 < a < 1$ and $b > 0$. However, in this case study, the estimates of the hyper-parameters for CBS2 were $a = 2.2$ and $b = 2793$ when using data set SYS2, and $a = 2.9$ and $b = 482$ when using data set Ckyjm, so the prior probability density functions are $\Gamma(2.2, 2793)$ and $\Gamma(2.2, 2793)$, which are not decreasing functions, so the results for the CBS2 scheme are not very good.

4.2. *Validation for DBSDF*

4.2.1. *Sources of the prior information.* The sources of the prior information are listed in Table 4. These were selected from two real failure data sets 'CSR1' and 'xrzmo', and represent the data for the last ten times between failures $T_1, T_2, \ldots, T_{10}$.

These data sets were collected from different processes of the reliability growth tests for some safety critical real-time control system software, and considered suitable for this case study. We assume that the number of test cases for each data set d is 100,000, so the sequence of empirical failure numbers after 100,000 test cases have been executed is

$$\{k_i\}_{i=1}^{10} = \{d/T_i\}_{i=1}^{10},$$

as listed in Table 4.

4.2.2. *SRDT index* (p_0, c, r). A feature of safety critical software is that the failure probability p_0 is between 10^{-5} and 10^{-3}. So, with this and properties of the data sets (for example, the range of value of the estimated failure probability) in mind, the index of failure probability p_0 was set to 10^{-3} for data set CSR1 and 10^{-4} for xrzmo. The

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 5. *The required number of SRDT test cases for each of the schemes for the CSR1 dataset*

r	DBS1	DBS2	DBSDF
0	4602	4347	3198
1	6635	5930	5231
2	8402	7452	6998
3	10041	8927	8637
4	11600	10366	10196
5	13104	11778	11699

confidence level c was set to 0.99 and the tolerance number for failure r was set variously to 0, 1, 2, 3, 4 and 5.

4.2.3. *Results and results analysis.* In this section we discuss the results of the case study for the CSR1 and xrzmo datasets:

— **Results and results analysis for CSR1:**
Given the prior information of the failure data set CSR1 listed in Table 4, the estimated values of the hyper-parameters of the prior probability density function for the DBS2 scheme can be calculated using the method proposed in Yang *et al.* (2004) to be $a = 3.6$ and $b = 5132$, so the prior probability density function for DBS2 is

$$f(p) = \frac{p^{2.6}(1-p)^{5131}}{B(3.6, 5132)}, \tag{35}$$

where

$$B(a,b) = \int_0^1 p^{a-1}(1-p)^{b-1}dp.$$

From Equation (35), the posterior probability density function of DBS2 is

$$f(p \mid r, n, a_0, b_0) = \text{Beta}\,(3.6 + r, 5132 + n - r). \tag{36}$$

Using the method presented in Equation (26), the estimated value of the hyper-parameter of the prior probability density function for the DBSDF scheme is $a = 1404$, so the prior probability density function for DBSDF is

$$f(p) = 1405(1-p)^{1404}. \tag{37}$$

Using the SRDT index $(0.001, 0.99, r)$ set above, we can use Equations (36), (37) and (30) to determine the number of SRDT test cases required for the DBS1, DBS2 and DBSDF schemes for data set CSR1 with r set to 0, 1, 2, 3, 4 and 5 – see Table 5 for the results.

Table 5 shows that for each r value (the tolerance of the number of failures), the required numbers of SRDT test cases calculated using the same SRDT index for the schemes with prior information (that is, DBS2 and DBSDF) are all smaller than those

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

Table 6. *The required number of SRDT test cases for each of the schemes for the xrzmo dataset*

r	DBS1	DBS2	DBSDF
0	46049	47340	44664
1	66380	67170	64995
2	84056	84638	82671
3	100447	100908	99062
4	116042	116422	114657
5	131080	131401	129695

calculated for the scheme without prior information (that is DBS1). The reductions in the number of SRDT test cases required are 10.3% for DBS2 and 15.5% for DBSDF. This suggests that the prior distribution of the failure probability can be more accurately described for the discrete BSRDT scheme with prior information, and this can significantly reduce the required number of SRDT test cases when valid prior information is available. On the other hand, when using the same prior information, the number of SRDT test cases required varies with different choices of prior distribution function.

— **Results and results analysis for xrzmo:**
In the same way as for the previous case, but using the xrzmo data set listed in Table 4 for the prior information for the failure, the estimated values of the hyper-parameters of the prior probability density function for the DBS2 scheme are $a = 1.1$ and $b = 1562$, so the corresponding prior probability density function is

$$f(p) = \frac{p^{0.1}(1-p)^{1561}}{B(1.1, 1563)}. \tag{38}$$

And the estimated value of the hyper-parameter of the prior probability density function for the DBSDF scheme is $a = 1385$, so the corresponding prior probability density function is

$$f(p) = 1386(1-p)^{1385}. \tag{39}$$

Using the SRDT index set above $(0.001, 0.99, r)$, we can use Equations (38), (39) and (30) to determine the number of SRDT test cases required for the DBS1, DBS2 and DBSDF schemes for the xrzmo data set with r set to 0, 1, 2, 3, 4 and 5 – see Table 6 for the results.

Table 6 shows that the required numbers of SRDT test cases calculated using the DBSDF scheme are again, for each value of r, all smaller than those calculated using the CBS1 and CBS2 schemes. However, the required numbers of test cases calculated using the CBS2 scheme are all larger than those calculated using the CBS1 scheme: specifically, they are larger by 3%, 1%, 0.6%, 0.5%, 0.3% and 0.2%, respectively. This means that when the selected prior probability density function is poor, the results for the Bayesian based scheme with prior information DBS2 are worse than for the

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

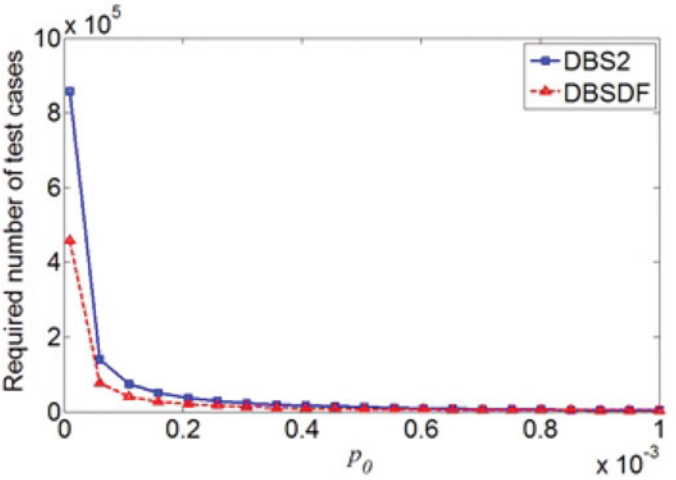

Fig. 5. (Colour online) Required numbers of test cases of CBS2 and CBSDF (CSR1)

scheme DBS1 without prior information. Thus, we have again shown that selecting an appropriate prior distribution for the failure probability for the failure data set is very significant in determining the effectiveness of the SRDT scheme.

4.2.4. *Analysis of the applicability of the DBSDF scheme.* To provide further evidence that DBSDF is a more suitable scheme for discrete highly reliable safety critical software (that is, with an index of failure probability ranging from 10^{-5} to 10^{-3}), we calculated the results for the DBS1, DBS2 and DBSDF schemes as the index of failure probability ranged from 10^{-5} to 10^{-3} with a step length of 4.95×10^{-5}.

For this test, we used prior information from the CSR1 and xrzmo data sets, and set the confidence level c to 0.99 and the tolerance for the number of failures r to 0.

Using the SRDT index set above, we calculated the required number of test cases using the DBS1, DBS2 and DBSDF schemes. We then plotted the required number of test cases for the DBS2 and DBSDF schemes in Figures 5 and 6, respectively, where the horizontal axis is the failure probability p_0 and the vertical axis is the required number of test cases. Finally, we calculated the reduction in the required number of SRDT test cases for the DBSDF and DBS2 schemes compared with the DBS1 scheme and plotted the results in Figures 7 and 8, respectively, with the same axes, so the line $y = 0$ can be viewed as the baseline DBS1 scheme. In this way, the performance of the DBSDF, DBS2 and DBS1 schemes can be evaluated by a visual comparison.

Summarising the results:

— Figures 5 and 6 show that:

(1) When the value of p_0 is increased from 10^{-5} to 10^{-3}, the required numbers of SRDT test cases for the DBS2 and DBSDF schemes show a decreasing trend, which suggests that a larger index of failure probability means a smaller required number of SRDT test cases for the scheme.

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

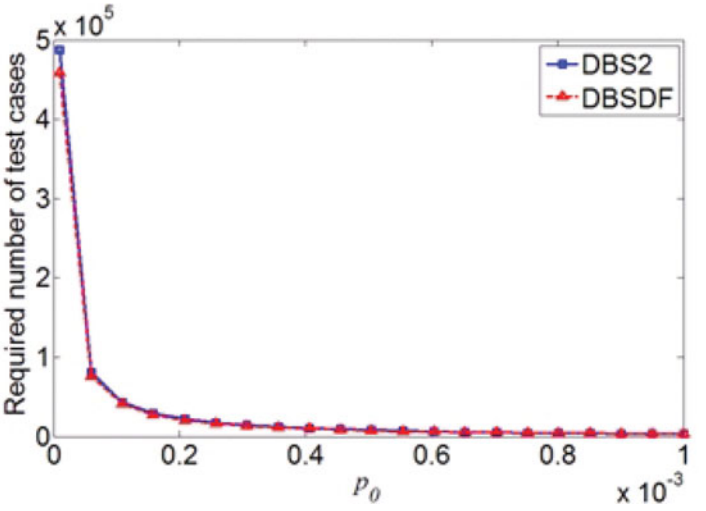

Fig. 6. (Colour online) Required numbers of test cases of CBS2 and CBSDF (xrzmo)

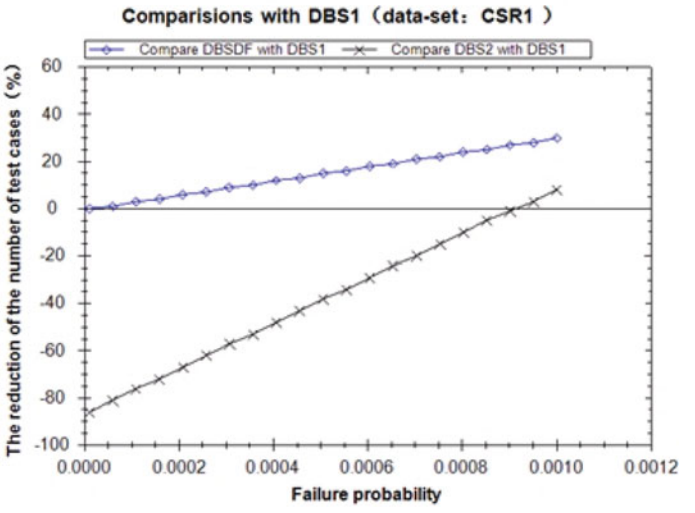

Fig. 7. (Colour online) Comparing CBSDF and CBS2 with CBS1 (CSR1)

(2) When the index of failure probability is small, specifically, when

$$10^{-5} < p_0 < 8 \times 10^{-5},$$

the required number of SRDT test cases for DBSDF will be significantly less than for DBS2. However, when the value of p_0 is increased gradually, the curve for DBSDF tends to be moderate, so the gap between the required number of SRDT test cases for DBSDF and for DBS2 decreases. This shows that DBSDF is more effective for discrete high reliability safety-critical software compared with DBS2.

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

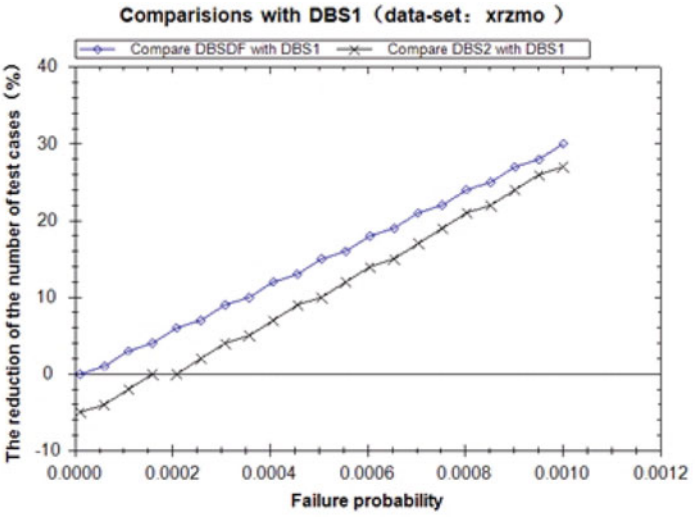

Fig. 8. (Colour online) Comparing CBSDF and CBS2 with CBS1 (xrzmo)

— Figures 7 and 8 show that:

(1) When $p_0 \in [10^{-5}, 10^{-3}]$, the curve for DBSDF is always above both the baseline of the DBS1 scheme without prior information and the DBS2 curve. In other words, the required number of SRDT test cases for DBSDF is always smaller than that for DBS1 and DBS2. This means that DBSDF is more effective than both DBS1 and DBS2 in terms of effectiveness and applicability.

(2) When the index of failure probability is close to 0.00001, the DBS2 scheme requires markedly more SRDT test cases than DBS1 does – see Figure 7. For example, when $p_0 = 0.00001$, the DBS2 scheme can require as much as 82% more effort than DBS1. This means that when the reliability index is high, the prior probability density function of failure probability for DBS2 is unsuitable for describing the distribution of the prior distribution of the failure probability, and it may lead to relatively poor results.

(3) However, when the reliability index is high, the required numbers of SRDT test cases for the schemes with prior information (DBSDF and DBS2) are significantly smaller than for the scheme DBS1 without prior information – see Figure 8. For example, when $p_0 = 0.001$, the required number of test cases for DBSDF is reduced by 31%, and for DBS2 it is reduced by 27%. This means that when valid prior information is available, the discrete BSRDT scheme with prior information can indeed substantially reduce the required number of SRDT test cases.

(4) When applied to discrete safety critical software, the DBSDF scheme is obviously superior to DBS2. Despite using prior information, DBS2 is sometimes worse than the DBS1 scheme without prior information. This is because the DBS2 scheme's prior probability density function is a Beta function Beta (a, b), which only becomes a decreasing function when $0 < a < 1$ and $b > 0$. However, the estimates of the DBS2 hyper-parameters in this case study are $a = 3.6$ and $b = 5,132$ for the

https://doi.org/10.1017/S0960129513000789 Published online by Cambridge University Press

CSR1 data set and $a = 1.1$ and $b = 1,562$ for the xrzmo data set, so the prior probability density functions are not decreasing functions, and the results for the DBS2 scheme are not very good.

5. Conclusions

The Bayesian SRDT scheme proposed by Littlewood selects the Gamma or Beta function as the prior distribution function for the failure rate or failure probability using a conjugate distribution. Since then, the research on Bayesian SRDT schemes has not focused much on the form of the prior distribution function. In the current paper we use decreasing functions to construct a prior distribution function that is more appropriate for describing the features of failure metrics. Based on this, we also propose a novel Bayesian-based SRDT scheme for safety critical software.

We carried out some case studies as part of our research. The experimental results show that with the same software reliability metric/index requirement, our proposed Bayesian-based SRDT schemes using a decreasing function (specifically, the DBSDF and CBSDF schemes) are more effective and applicable than other current Bayesian-based schemes. Our results have considerable engineering significance and practical value in the application of software reliability demonstration testing, especially for safety critical software with a high reliability requirement since it helps reduce the test effort required and improves test efficiency.

References

Cukic, B. and Chakravarthy, D. (2000) Bayesian framework for reliability assurance of a deployed safety critical system. In: *Proceedings of the 5th IEEE International Symposium on High Assurance Systems Engineering* 321–329.

Department of Defense (1996) Reliability test methods, plans, and environments for engineering, development qualification, and production. *Military Handbook MIL-HDBK-781A*, Department of Defense, United States of America.

Dey, D. K. and Rao, C. R. (2005) *Handbook of Statistics 25: Bayesian Thinking: Modeling and Computation*, Elsevier.

Han, M. (2004) *The estimation of reliability parameters without failure data* (in Chinese), Chinese Statistic Press 66–67.

Lindley, D. V. and Smith, A. F. M. (1972) Bayes estimation for the linear model. *Journal of the Royal Statistical Society* **34** (1) 1–41.

Littlewood, B. and David, W. (1997) Some conservative stopping rules for the operational testing of safety critical software. *IEEE Transactions on Software Engineering* **23** (11) 673–683.

Lyu, M. R. (1996) *Handbook of Software Reliability Engineering*, McGraw Hill.

Miller, K. W. *et al.* (1992) Estimating the probability of failure when testing reveals no failures. *IEEE Transactions on Software Engineering* **18** (1) 33–43.

Qin, Z. and Lei, H. (2004) Research on safety testing and evaluation technology of safety critical software (in Chinese). *Chinese journal of computers* **27** (4) 442–451.

Qin, Z., Chen, H. and Shi, Y. (2008) Reliability demonstration testing method for safety-critical embedded applications software. *Proceedings of International Conference on Embedded Software and Systems* 481–487.

Qin, Z., Lei, H., Sang, N. and Xiong, G. (2005) Reliability demonstration testing method for continuous execution software (in Chinese). *Computer Science* **32** (6) 202–205.

Qin, Z., Lei, H., Sang, N., Xiong, G. and Gu, Y. (2005) Study on the reliability demonstration testing method for safety-critical software. *ACTA Aeronautica et Astronautica Sinica* **26** (3) 334–339.

Rahrouh, M. N. (2005) Bayesian zero-failure reliability demonstration, University of Durham.

Tal O., Bendell, A. and McCollin, C. (2000) A comparison of methods for calculating the duration of software reliability demonstration testing, particularly for safety-critical systems. *Quality and Reliability Engineering International* **16** (1) 59–62.

Tal, O., McCollin, C. and Bendell, T. (2001) Reliability demonstration for safety-critical systems. *IEEE Transactions on Reliability* **50** (2) 194–204.

Thayer, T. A., Lipow, M. and Nelson, E. C. (1978) *Software reliability-TRW Series of Software Technology*, North-Holland Publishing.

Yang, S., Xiong, G., Sang, N. and Wu, X. (2004) Research on safety evaluation of high dependable software. *Computer Engineering and Design* **25** (2) 161–166.