Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 6, October 2017, pp. 832-843
DOI: 10.23940/ijpe.17.06.p5.832843

A Distribution-Level Combinational Model to Improve Reliability Prediction Accuracy

Wenjun Xie[a], Haiyan Sun[a], Lu Zhang[a], Ji Wu[b,*]

[a] *The LIMB of the Ministry of Education, School of Mathematics and Systems Science, Beihang University, Beijing, 100191, China*
[b]*School of Computer Science and Engineering, Beihang University, Beijing, 100191, China*

Abstract

Single reliability growth model usually only captures partial knowledge of a failure process. A combinational model tries to capture more knowledge by integrating two or more reliability growth models. Unlike the existing linear combinational models that simply adds up the weighted results by G-O, M-O and L-V model, this paper proposes the combinational model from G-O and S-Shaped model, at the level of failure distribution to reduce fitting errors and to maintain the mathematical properties of non-homogenous Poisson process. To evaluate the effectiveness of the proposed model, we use the failure data sets (21 projects) available in public sources. Ten out of the twenty-one projects, which pass the distribution test and have feasible solutions in parameter estimation, are selected to conduct experiments. We use mean squared error (MSE) to evaluate the historical predictive validity. The results show that our model is consistently stable and has lower MSE. It reduces 51.3% MSE of G-O, 67.2% MSE of S-Shaped, and over 56% MSE of the three linear combinational models in average. The proposed model tends to have a larger estimation of the expected number of failures, which can overcome the under estimation by G-O and S-Shaped model in some degree.

Keywords: software reliability growth model; failure distribution; combinational model; G-O model; S-Shaped model

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

1. Introduction

Reliability is a probability metric that a software system runs successfully in a given environment for a given time duration. As software system becomes ubiquitous, unreliable software might cause serious, even catastrophic results on human society. Therefore, how to assure and assess software reliability has become an important topic from time to time. Software testing might be one of the most popular techniques to assure software reliability. From engineering perspective, testers would use any kind of testing approaches and tools at hand to uncover as many failures as possible. Due to the complexity of software failing process, reliability growth model usually treats it as a stochastic process based on some necessary assumptions (e.g. Non-Homogeneous Poisson Process [5], NHPP, assumption).

For a given specific software failure data set, engineers can find several applicable models, with different statistical fitting errors. We use the mean squared error, *MSE* [24] in this paper to evaluate fitting effects. Engineers usually select the model with minimal *MSE*. Even if the selected model performs best at a certain project, as new project comes, there is no guarantee that it still has the minimal *MSE*. One of the key reasons is that the complicated failure process of software provides only partial knowledge of the process captured by single reliability model. Therefore, it motivates this paper to work on combinational reliability growth model based on candidate models in a theoretic sound way to reduce fitting errors, thus improve prediction accuracy.

* Corresponding author.
E-mail address: wuji@buaa.edu.cn.

In general, there are two types of reliability growth models, finite model and infinite model [19] (pages: 395~424). Finite model assumes that any software has a finite number of faults, while infinite model assumes infinite number of faults in software. The Poisson process based finite model treats software failure process as a stochastic Poisson process. Goel-Okumoto(G-O) model [5] and S-Shaped model [27], as typical representative Poisson process based finite models, are widely applied in practice [7]. We propose the combinational model by combining the exponential distribution (employed by G-O model) and gamma distribution (employed by S-Shaped model). Users can apply our model if a given data set passes the tests of exponential or gamma distribution. To assess the effectiveness of the proposed model, we use 21 data sets available in public sources to conduct experiment, and compare our model with G-O, S-Shaped, and three linear combinational models (ELC [13], MLC [13] and ULC [14]) in terms of *MSE*. We also discuss the parameter estimation results in the experiment.

This paper is structured into six sections. The research problem is formally defined in section 2, and we introduce the related work in section 3. We propose the combinational model in section 4 based on the problem defined, and in section 5 we introduce the experiment results and discussions. Finally, we conclude this paper in section 6.

2. Problem Definition

The objective of this paper is to derive a combinational reliability growth model by integrating the exponential distribution function of G-O model and the gamma distribution function of S-Shaped model. Before introducing the combinational model, we first define the problem.

Software reliability growth model (SRGM) is a stochastic model to represent fault detection processes in testing phases. A SRGM usually puts some appropriate stochastic assumptions on the distribution of failure occurrences and software debugging process [3]. In general, the process of detecting and fixing software failures can be specified in Markov process [17]. NHPP based SRGM is very popular due to its mathematical tractability [20,21,22], and there have been a number of NHPP-based SRGMs proposed, such as G-O and S-Shaped model.

Let $N(t)$ be the number of failures uncovered to time t, then $N(0)=0$ by definition. Let $P_n(t)$ be the probability of n failures uncovered to time t, $P_n(t)=P\{N(t)=n\}$, then $P_0(0)=1$ by definition. According to the NHPP assumptions [19], we know that (a) $N(t)$ is a monotonic increasing function, i.e. $N(t+\Delta t)\geq N(t)$, and $N(t+\Delta t)-N(t)$ is independent of $N(t)$; (b) the probability of observing one failure in the duration $(t,t+\Delta t]$ is $\lambda(t)\Delta t+o(\Delta t)$, where $\lambda(t)$ is the failure intensity, while the probability to observe two or more failures in the duration $(t,t+\Delta t]$ is $o(\Delta t)$. The mean failure function, $m(t)=\int_0^t \lambda(s)ds$, defines the mean value of a Poisson process.

We can define the research problem of this paper formally: how to derive a combinational mean failure function $m_D(t)=\omega * m_G(t)+(1-\omega)*m_S(t)$ to propose a combinational reliability growth model to capture the comprehensive distribution knowledge of a failure process, where ω is the weight, $m_G(t)$ is the mean function of G-O model and $m_S(t)$ is the mean function of S-Shaped model.

3. Related Work

There are over 40 kinds of software reliability models published [17], and no single model can be applied in all situations [2,11,17]. Among the published models, G-O and S-Shaped are the two widely applied ones in practice. In general, G-O model reflects the pattern of initially increasing failure density, and decreasing failure density later on [17]. Ramasamy proposes the shifted Weibull distribution to extend Goel generalized model to sustain the stableness with fluctuations in the time between failures [22]. Though a model achieved good result in a given failure data sample, it still cannot assure the effectiveness when a new data set added [13]. By considering the differences of testing phase and operation phase, fault detection rate measured from operation phase is used to improve the effectiveness of G-O model [28]. Instead of extending G-O or S-Shaped model, we combine them.

Many efforts have been spent to apply reliability growth model in practice. We focus on combining models to improve the effectiveness of reliability assessment. Before conducting reliability assessment, one should select a reliability growth model at first by examining the assumptions that a candidate model makes about the development method and environment [12], or by evaluating various criteria of fitting effects [24], or by exploring the characteristics of failure data [1].

Sometimes, it is hard to select an optimal growth model with limited available failure data [2]. Combining several appropriate models is an out-of-the-box approach to explore given failure data better [12,14,23]. Among the combinational models, ELC [13], integrates G-O, M-O [18] and L-V model [10] linearly with equal weights. The basic idea of ELC model is to cancel out in their biased prediction since G-O tends to produce under estimation, L-V model tends to have over estimation, and M-O model is quite flexible [12,13]. There are several weighted approaches proposed, including MLC that uses the median of results among the three models [13], ULC model that uses unequal weights [14], and DLC model that works with dynamic weights according to a given failure data [12]. In fact, DLC model is just a framework. To apply this kind of model, one needs to select appropriate base models to combine at first, and then to decide the corresponding weights dynamically, such as applying neural network approach to estimate the weights [25]. Jinhee uses decision tree to select base models and to decide the weights according to the prediction accuracies of the selected base models [21].

Since DISCOR combines G-O and S-Shaped model at the level of failure distribution, we would not compare it with any DLC model. We compare it with ELC, MLC and ULC in terms of *MSE*, because these three models are specific and all employ G-O model. These existing combinational models use the linear combination approach at the level of the final fitting results from their base models respectively. While DISCOR model takes the combination approach at the level of distribution function, which is fundamentally different, DISCOR model sustains the properties of NHPP such that one can study with it in mathematical ways strictly.

4. DISCOR Model

We propose the Distribution-level Combinational Reliability (DISCOR) model by integrating the mean failure functions of G-O model and S-Shaped model to characterize the failure distribution of a real data set better.

According to the NHPP assumptions introduced in the Section 2, the transition function $P_{ij}(t,\Delta t)$ of the stochastic process $N(t)$ is defined in (1), as introduced in [19] (pages 255~257). It specifies the probability of uncovering one or more failures (*j*-*i*) in the duration $(t,t+\Delta t]$ given that the number of failures uncovered to time *t* is *i*.

$$P_{ij}(t,\Delta t)=\begin{cases}1-\lambda(t)\Delta t+o(\Delta t), & j=i\\ \lambda(t)\Delta t, & j=i+1\\ o(\Delta t), & otherwise\end{cases}. \tag{1}$$

We can get the following equation (2) by taking the above transition function of the process into the equation $P_n(t+\Delta t)=\sum_i P_{i,n}(t,\Delta t)P_i(t)$, which is derived according to the Markovian property.

$$P_n(t+\Delta t)=[1-\lambda(t)\Delta t]P_n(t)+\lambda(t)\Delta t P_{n-1}(t)+o(\Delta t) \tag{2}$$

Then we have the following equation (3).

$$P_n(t+\Delta t)-P_n(t)=-\lambda(t)\Delta t P_n(t)+\lambda(t)\Delta t P_{n-1}(t)+o(\Delta t) \tag{3}$$

By dividing the both sides of equation (3) with Δt and let $\Delta t\to 0$, we have the following differential equation in (4).

$$\frac{\partial P_n(t)}{\partial t}=\lambda(t)[P_{n-1}(t)-P_n(t)] \tag{4}$$

By using generating function approach, we define the function $G(t,z)$ in (5). Then we have the partial differential equation of the generating function in (6).

$$G(t,z)=\sum_{n=0}^{\infty}P_n(t)z^n \tag{5}$$

$$\frac{\partial G(t,z)}{\partial t}=\lambda(t)(z-1)G(t,z) \tag{6}$$

We can solve this partial differential equation to get $G(t,z)$, as shown in (7). By comparing equation (5) and (7), we can easily derive $P_n(t)$ in (8).

$$\begin{aligned} G(t,z) &= e^{-(z-1)\int_0^t \lambda(x)dx} \\ &= e^{-zm(t)}e^{m(t)} \\ &= \sum_{n=0}^{\infty}\frac{[m(t)]^n}{n!}e^{-m(t)}z^n \end{aligned} \tag{7}$$

$$P_n(t)=\frac{[m(t)]^n}{n!}\exp[-m(t)] \tag{8}$$

Since many factors contribute to the expected number of failures to be uncovered in software, we define it as a random variable with mean value of $N(t)$ for the finite NHPP model [17]. The structure of $P_n(t)$ clearly shows that $N(t)$ conforms to a Poisson process with the mean function $m(t)$. By definition, we have $m(t)=NF(t)$, where $F(t)$ is the distribution function of time to failure. Then we can refine $P_n(t)$ as:

$$P_n(t)=\frac{[NF(t)]^n}{n!}\exp[-NF(t)]. \tag{9}$$

$F(t)$ can be derived from the definition of finite NHPP model. For example, G-O model has $F(t)=1-e^{-\beta t}$, which is an exponential distribution; while S-Shaped model has $F(t)=1-(1+bt)e^{-bt}$, which is a gamma distribution [5,27,28]. The better $F(t)$ fits to the distribution of a given failure data set, the less fitting error *MSE* the model $P_n(t)$ will have. The distribution of time to failure of real world software project can be very complicated, and is affected by many factors. Single distribution function might only capture partial knowledge. Therefore, we propose the combinational distribution function based on G-O model and S-Shaped model, which are popularly applied in engineering practices.

Let the distribution of time to failure of DISCOR model be $F_D(t)$, its density function is therefore $f_D(t)$, and $F_D(t)=\int f_D(s)ds$, therefore $\lambda_D(t)=N*f_D(t)$ since $m(t)=NF(t)$. Then we have the following expression (10), where $f_G(t)$ is the density function for G-O model and $f_S(t)$ is the density function for S-Shaped model:

$$f_D(t)=\omega*f_G(t)+(1-\omega)*f_S(t). \tag{10}$$

Now we can refine the DISCOR model with $F_D(t)=\omega*\int f_G(s)ds+(1-\omega)*\int f_S(s)ds$. Finally, we have the mean function $m_D(t)$ of DISCOR model, where N,ω,β and b are the four parameters to estimate. ω is a new parameter for weighting, and others constitute two pairs of parameters, which are (N, β) in G-O and (N and b) in S-Shaped.

$$m_D(t)=\omega*N\left(1-e^{-\beta t}\right)+(1-\omega)*N\left(1-(1+bt)e^{-bt}\right) \tag{11}$$

We apply the widely used Maximum Likelihood Estimation (MLE) to estimate the parameters in this paper. The likelihood function we use is defined in (12), where $t_1, t_2, ..., t_{m_e}$ are the time points when failure 1, 2, …, m_e were observed respectively (pages 318~324) [19].

$$L(N, \omega, \beta, b) = [\prod_{i=1}^{m_e} \lambda_D(t_i)] \exp(-m_D(t_e)) \tag{12}$$

To solve this likelihood function, we at first set initial values of parameters from which to search for the feasible solutions. *N* is the most important parameter in DISCOR, we will discuss how the initial value affects the final estimation and *MSE*.

5. Experiment and Discussion

This section introduces the experiment setting, experiment results, and discussions.

5.1. Experiment Setting

To evaluate the effectiveness of DISCOR model, we choose 16 data sets collected in Bell labs and published by NASA DACS [16], 3 ones (PA1~PA3) published by Abdel-Ghaly [1], and 2 ones (PSt1, PSt2) published by Mullen [15], which are shown in Table 1. The data sets from NASA DACS are from 16 real software projects with size (number of delivered object code instructions) ranging from 5.7 thousand to 2445 thousand. In addition, the data sets published by Mullen are from an operating system project with over 1 million lines of code. However, the data sets published by Abdel-Ghaly do not provide the size information of projects. Since DISCOR model combines G-O and S-Shaped model, we conduct distribution tests of exponential and gamma distribution at first to select the appropriate data sets for experiment from the 21 ones.

We use the criteria of historical predictive validity to evaluate DISCOR model by comparing the retrospective predictions with the real failure data set [7]. We compare DISCOR model with G-O, S-Shaped and the three linear combinational models in terms of *MSE*. All data points are used to estimate parameters, and we select eight observational data points at $ot_1, ot_2, ..., ot_8$ in time axis from 20% (i.e., ot_1) to 90% (i.e., ot_8) of the total time length with the step of 10% of the length. Then we calculate the deviation from the estimation $\hat{m}_D(ot_i)$ and the observed $m_D(ot_i)$ correspondingly. The eight deviations are then summarized to calculate the *MSE*:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} [\hat{m}_D(ot_i) - m_D(ot_i)]^2 \tag{13}$$

5.2. Experiment Results

At the first step, we conducted distribution tests for all the data sets of 21 projects. If a distribution test has *p-value* greater than 0.05, we say the corresponding project data passes the test. As shown in the Table 1, where PID refers to project ID, *p-v(e)* is the *p-value* of exponential distribution test and *p-v(g)* is the *p-value* for gamma distribution test. All the 12 projects that pass at least one test are highlighted in green.

Table 1. Distribution test results for all the 21 projects

PID	*p-v(e)*	*p-v(g)*	PID	*p-v(e)*	*p-v(g)*	PID	*p-v(e)*	*p-v(g)*
P1	0.2173	0.1457	P14c	0.0039	0.7918	PSS1C	0.00063	<0.0001
P2	0.4560	0.4340	P17	0.4429	0.9492	PSS2	<0.0001	0.00108
P3	0.0261	0.3928	P27	0.0791	0.0363	PSS3	<0.0001	0.01091
P4	0.7557	0.2343	P40	<0.0001	0.00678	PSS4	<0.0001	0.00787
P5	<0.0001	<0.0001	PSS1A	0.00095	0.19352	PA1	0.6000	0.8794
P6	0.0011	0.0023	PSS1B	<0.0001	<0.0001	PA2	0.6106	0.5860
PSt1	0.0019	<0.0001	PSt2	0.0786	0.0414	PA3	0.0338	0.0816

With the 12 selected failure data sets, we then conducted parameter estimation by using the likelihood function (12) introduced in section III. Please note that we use the least square method to estimate the parameters of the linear

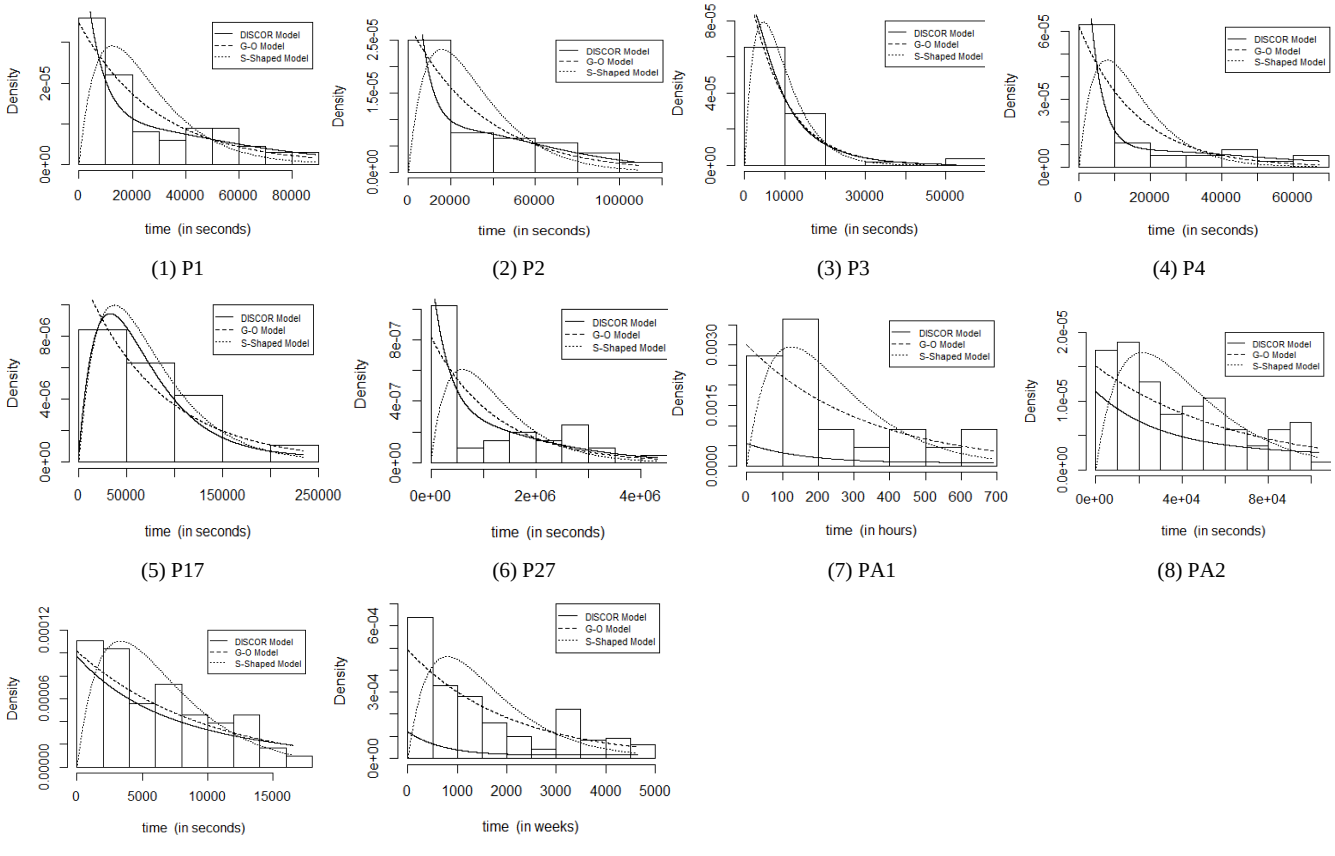
combinational models since they compose of L-V model that cannot use MLE [4]. There is no feasible solution for the four parameters in project P14c and PSS1A. Therefore, we have 10 projects to evaluate DISCOR model. Table 2 reports the basic statistics for the 10 projects. There are four types of projects and the type of PA2 and PA3 remains unknown.

Table 2. Basic statistics of the ten selected projects

PID	*Type*	*NoF*	*Duration*	*size*
P1	RT&C	136	91208(seconds)	21700
P2	RT&C	54	118006(seconds)	27700
P3	RT&C	38	77537(seconds)	23400
P4	RT&C	53	66642(seconds)	33500
P17	Military	38	28200(seconds)	>100000
P27	Military	41	6477878(seconds)	61900
PA1	NWS	22	40800(seconds)	Unknown
PA2	Unknown	86	103284(seconds)	Unknown
PA3	Unknown	207	16553(seconds)	Unknown
PSt2	OS	200	4628(weeks)	>100,0000

Type RT&C refers to real time and communication. Type Military means a project serves for some military objective. Type NWS means network system, and OS refers to operating system. *NoF* is the number of failures observed (i.e. provided in data set), duration is the time length of tests conducted to collect failure data, and size is the number of delivered object code instructions.

As suggested by Barbara Kitchenham, density plot is a robust statistical method to visualize the distribution of data set [9,10]. We at first draw the density as histograms and the distribution fitting lines of DISCOR, G-O and S-Shaped models, as shown in Figure 1.



(1) P1 (2) P2 (3) P3 (4) P4

(5) P17 (6) P27 (7) PA1 (8) PA2

(9) PA3 (10) PSt2

Figure 1. Density plots of failure data and the fitting effects.

Since the linear combination models are not relevant with failure distribution, we only compare DISCOR model with G-O and S-Shaped model in the above density plots. We can clearly read the failure trend of each project along time,

especially the tail of each plot. Project P1, P2, P3, P4 and P17 have quite long and thin tails, while PA1, PA2 and PSt2 have relative short and thick tails. We find in Figure 1 that ***DISCOR model fits to the failure trends better than the others do***.

We further evaluate DISCOR model in terms of *MSE*, as shown in Table 3. We can directly find that DISCOR model has almost the minimal *MSE* in all the projects except P17, in which MSE_D is small but a little larger than the MSE_S. We define the measure Relative Error Reduced, *RER (D, X)* to compare DISCOR and other models. *RER (D, X)* is computed as the percentage of *MSE* decreased in the X model than the D model, where D refers to DISCOR model:

$$RER(D, X) = \frac{MSE_X - MSE_D}{MSE_X} * 100\% \cdot \tag{14}$$

We use the first letter to abbreviate the models compared in Table 3. For example, G refers to G-O model, and U refers to ULC model. We can find there are 44 *RER* measures (among the 50 ones) larger than 30% in the ten projects. That means DISCOR consistently reduces over 30% *MSE* of the other models. There are 22 *RER* measures even larger than 80%.

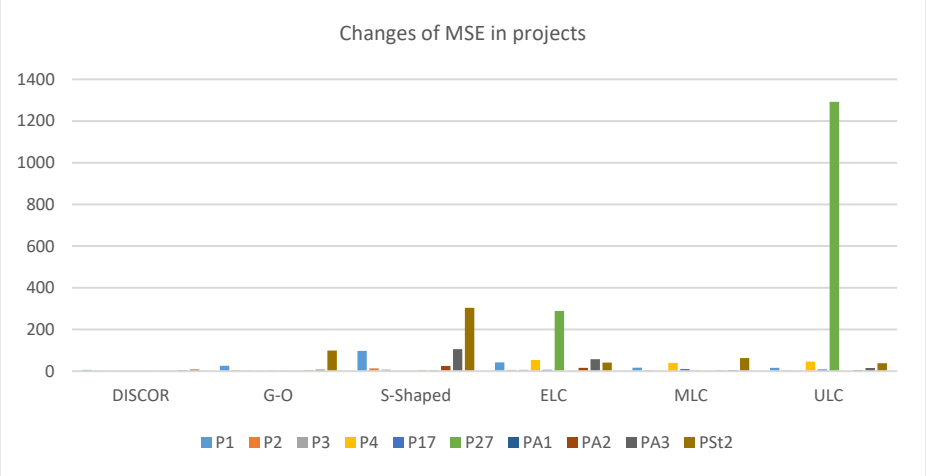


Figure 2. Changes of *MSE* in all projects

Besides the reduced relative error, we conduct the stability analysis of *MSE* changes in all the projects, as shown in Figure 2. We can find that DISCOR, G-O and MLC are more stable than the others are. To compare DISCOR, G-O and MLC in further, we zoom in on the chart to show the *MSE* changes of these three models (in Figure 3).

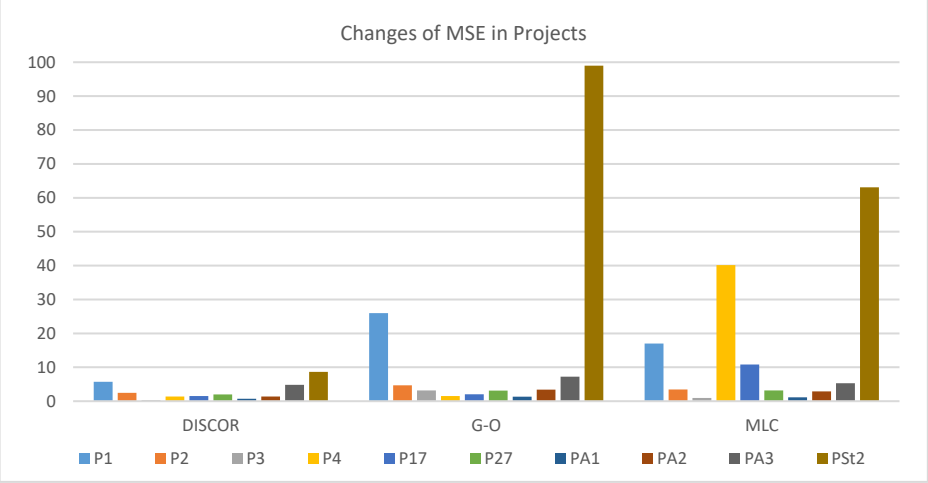


Figure 3. *MSE* changes of DISCOR, G-O and MLC in all projects

We can find that DISCOR is the most stable one according to the changes of *MSE* in the ten projects in Figure 3. All the *MSE* values of DISCOR are less than 10. The *MSE* of G-O in P1 is over 25, the *MSE* of G-O in PSt2 is over 95 even, and four *MSE* values of MLC are over 10, which are the ones in P1, P4, P17 and PSt2. By the way, these three models consistently have the largest *MSE* values at PSt2. Based on the *RER* measures and the stability analysis of *MSE* changes in projects, we conclude that ***DISCOR does not only reduce a significant percentage of MSE against the other models, also maintains the most stable MSE change pattern in all the projects***.

Table 3. *MSE* of applied models and the significance of failure distribution test

PID	*DISCOR*	*G-O*	*S-Shaped*	*ELC*	*MLC*	*ULC*	*RER(D,G)*	*RER(D,S)*	*RER(D,E)*	*RER(D,M)*	*RER(D,U)*
P1	5.744	25.991	97.225	41.947	17.0232	16.3605	77.9%	94.1%	86.3%	66.3%	64.9%
P2	2.468	4.697	13.780	6.022	3.4555	3.6232	47.4%	82.1%	59.0%	28.6%	31.9%
P3	0.299	3.182	7.871	7.208	0.9604	2.5797	90.6%	96.2%	95.9%	68.9%	88.4%
P4	1.398	1.527	1.635	53.845	40.1649	46.1831	8.4%	14.5%	97.4%	96.5%	97.0%
P17	1.530	2.023	1.137	6.633	10.8100	8.4508	24.3%	-34.6%	76.9%	85.8%	81.9%
P27	2.006	3.145	4.819	288.812	3.184	1292.379	36.2%	58.4%	99.3%	37.0%	99.8%
PA1	0.733	1.309	2.872	1.185	1.1575	1.1654	44.0%	74.5%	38.1%	36.7%	37.1%
PA2	1.375	3.404	25.358	16.417	2.8865	5.0222	59.6%	94.6%	91.6%	52.4%	72.6%
PA3	4.841	7.244	106.159	57.613	5.2901	14.9876	33.2%	95.4%	91.6%	8.5%	67.7%
PSt2	8.634	98.983	304.112	41.747	63.0910	38.3881	91.3%	97.2%	79.3%	86.3%	77.5%

5.3. Discussions

Based on the experiment results reported above, we would discuss the following three questions in this section:

- How does the setting of initial *N* affect *MSE* and the estimation of *N*?
- What are the differences between *N* estimations and the observed numbers of failures in DISCOR, G-O and S-Shaped model?
- What is the reason that DISCOR has a little larger *MSE* than S-Shaped model in P17 project?

5.3.1. Discussion 1: How does the setting of initial N affect MSE and the estimation of N?

The setting of initial value of a parameter usually has effects on the estimation result of the parameter. It could happen that we cannot find the feasible estimation of a parameter when applying MLE in some cases such as P14c and PSS1A on our experiment. In general, the selection of initial value of parameter might affect the effectiveness of a reliability growth model eventually [8].

N, the expected total number of failures uncovered in the whole life of software, is the most important parameter of NHPP based reliability growth model. To discuss this question, we conducted 20 repetitive runs with different initial values for each of the projects to estimate the parameter *N*. The results are reported in Table 4, where S is statistics (*Mn* for mean, *SD* for standard deviation); MSE_D, MSE_G and MSE_S are the *MSE* measures for DISCOR, G-O and S-Shaped model respectively; N_D, N_G and N_S are the estimations of *N* for DISCOR, G-O and S-Shaped model respectively. The standard deviation reflects the changes of *MSE* values and *N* estimations of the runs with different initial values.

We can find in Table 4 that most of the *SD* measures of *MSE* in the three models are very small (less than 1.0). That means, the variations of *MSE* with different settings of initial *N* are well controlled in all the projects. In particular, we find that DISCOR has less *SD* than G-O and S-Shaped model in most of the projects. That means ***DISCOR is less sensitive to the selection of initial values when estimating model parameters***. Please note that some initial values lead to the divergent likelihood function in project P3, P4, P17 and P27. In practice, users need to have several runs of parameter estimation with different initial values to avoid this situation.

Since different initial *N* values might lead to the different final estimations, and thus lead to changes of *MSE*, we are interested in checking whether there is statistical correlation between the *SD* of *MSE* and the *SD* of *N* estimation. We conducted the non-parametric Spearman test [6]; the result shows no significant correlation between them since the *p-values* are all larger than 0.05 (0.1839 for DISCOR and G-O, 0.0897 for S-Shaped). Since we only have 10 projects in the experiment, it might not guarantee the valid statistical test. We further define the ratio of *MSE* variation to *N* variation (*ROV*) in (15) to evaluate the change of *MSE* caused by the change of *N* estimation. As shown in the Table 4, ROV_D, ROV_G, ROV_S are the *ROV* measures for DISCOR, G-O and S-Shaped model respectively.

$$ROV = \frac{SD(MSE)}{SD(N)} \tag{15}$$

The measures of ROV_D, ROV_G and ROV_S in all the 10 projects are illustrated in the following Figure 4. We can find that DISCOR model has stably smaller *ROV* than G-O and S-Shaped model. That means ***DISCOR model is more robust to the different initial values of N***. The result also reminds the users of G-O model and S-Shaped model to select the initial value of *N* during parameter estimation carefully.

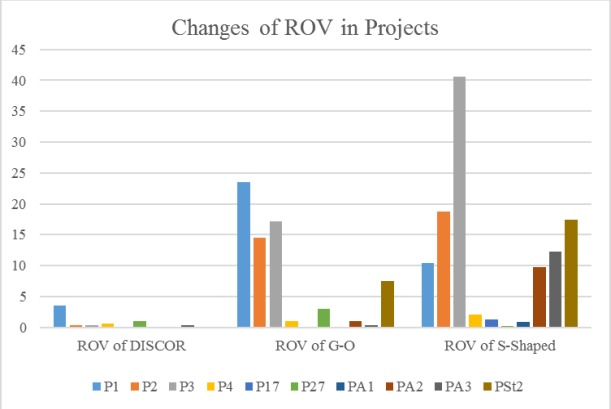


Figure 4. *ROV* changes of DISCOR, G-O and S-Shaped in all projects

5.3.2. Discussion 2: What are the differences between N estimations and the observed numbers of failures in DISCOR, G-O and S-Shaped model?

Different reliability growth model usually has different estimation of *N*. It was reported that G-O and S-Shaped model tend to have under estimation [18]. It is good to see in the Table 4 that DISCOR has larger estimated *N* than G-O and S-Shaped models in most of the projects, though we cannot conclude whether DISCOR tends to have under or over estimation in this paper because we do not have any field failure data regarding to the data sets from public. We would discuss the differences between final estimations of *N* and the observed numbers of failures (*NoF*) by the DISCOR, G-O and S-Shaped model. We define the measure *RDN* as the ratio of the number of failures expected to uncover in the future and *NoF*, as shown in Table 4, where RDN_D, RDN_G and RDN_S are for DISCOR, G-O and S-Shaped model respectively.

$$RDN = \frac{N - NoF}{NoF} * 100\% \tag{16}$$

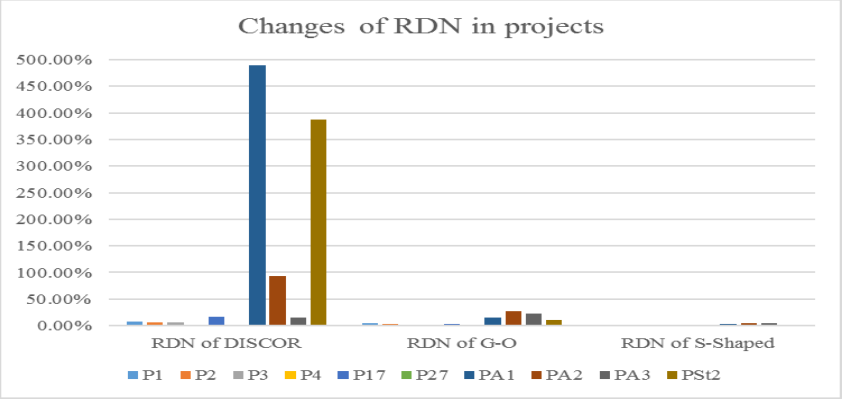


Figure 5. *RDN* changes of DISCOR, G-O and S-Shaped in all projects

As shown in Figure 5, S-Shaped model has the smallest *RDN* measures in all projects, while DISCOR has the biggest *RDN* measures in most of the projects. Since DISCOR model utilizes much more information in a given data set, it tends to be more sensitive to have larger changes of *RDN*, as we find in Figure 5. In project P4 and P27, the three models all agree that it is almost impossible to find one more failure in the future since the *N* estimation is nearly the same as *NoF*. We can find that DISCOR model has very high *RDN* in project PA1, PA2 and PSt2. When we look at the density plots (Figure 1), we find these three projects all have short and thick tails. It usually means that the tests to uncover failures terminated too quickly, thus more effort on testing is expected. Based on the observations and analyses, we conclude that ***DISCOR model***

tends to have larger estimation of N than G-O and S-Shaped model, and *RDN* can be used as an indicator of how much more effort should be spent on tests compared with the effort spent before to improve reliability in further.

Table 4. Changes of *MSE* in repetitive parameter estimations

PID	*NoF*	*S*	MSE_D	MSE_G	MSE_S	N_D	N_G	N_S	ROV_D	ROV_G	ROV_S	RDN_D	RDN_G	RDN_S
P1	136	Mn	5.846	25.419	97.104	146.7	141.9	136.8	3.519	23.568	10.398	7.87%	4.34%	0.59%
		SD	0.366	1.037	1.934	0.104	0.044	0.186						
P2	54	Mn	2.459	4.517	13.659	57.1	56	54.3	0.360	14.455	18.727	5.74%	3.70%	0.56%
		SD	0.009	0.318	0.206	0.025	0.022	0.011						
P3	38	Mn	0.301	3.257	7.883	40.1	38.3	38.01	0.333	17.167	40.625	5.53%	0.79%	0.03%
		SD	0.004	0.206	0.325	0.012	0.012	0.008						
P4	52	Mn	1.403	1.518	1.701	52.05	52.03	52.01	0.600	1.000	2.071	0.10%	0.06%	0.02%
		SD	0.015	0.014	0.029	0.025	0.014	0.014						
P17	38	Mn	1.654	2.023	1.149	44.07	39.25	38.16	0.146	0.042	1.300	15.97%	3.29%	0.42%
		SD	0.426	0.001	0.013	2.925	0.024	0.01						
P27	41	Mn	2.119	2.892	4.771	41.21	41.30	41.01	0.992	2.955	0.222	0.51%	0.73%	0.02%
		SD	0.129	0.328	0.184	0.13	0.111	0.829						
PA1	22	Mn	0.731	1.308	2.873	129.65	25.18	22.58	0.001	0.111	0.833	489.32%	14.45%	2.64%
		SD	0.004	0.0001	0.0005	3.253	0.0009	0.0006						
PA2	86	Mn	1.392	3.446	25.277	166.06	109.12	90.33	0.018	1.044	9.758	93.09%	26.88%	5.03%
		SD	0.06	0.141	0.322	3.275	0.135	0.033						
PA3	207	Mn	4.915	7.215	105.917	238.64	254.72	215.98	0.410	0.397	12.239	15.29%	23.05%	4.34%
		SD	0.167	0.095	1.383	0.407	0.239	0.113						
PSt2	200	Mn	8.800	99.542	303.933	975.188	222.904	204.062	0.001	7.522	17.422	387.59%	11.45%	2.03%
		SD	0.386	1.572	1.899	335.893	0.209	0.109						

5.3.3. Discussion 3: What is the reason that DISCOR has a little larger MSE than S-Shaped model in P17 project?

We can find in Table 3 that the *MSE* of DISCOR is a little larger than S-Shaped model only in project P17, though the *MSE* of DISCOR is already very small (1.530). Table 4 shows that DISCOR has larger *RDN* and modest *ROV* in P17. When we look at the data points in P17, we find that the time duration to detect the 37th failure is 79500 seconds, while the duration to detect the last (38th) one is 9000 seconds. After a new failure (the 37th) was uncovered in a very long duration (almost 13 times of the mean duration 6150), then quickly another failure (the 38th) was uncovered in 9000 seconds (1.46 times of the mean duration). That means there need much more testing time to stabilize the failure process since more failures would be expected to uncover in the future according to the dynamics of process [25,26].

We can see that the final *N* estimation in DISCOR model is 44.07, which means that DISCOR model believes there should have 6 more failures (=44.07-38) to uncover in the future since 38 failures are presented in the given data set. At the same time, we find that the final *N* estimations by G-O model and S-Shaped model are 39.25 and 38.16 respectively. Therefore, DISCOR overcomes the quick termination of testing in project P17 by producing a larger *N* estimation. Because the *MSE* of DISCOR in P17 is already very small (1.530), and *RER (D, S)* is also quite small (-34.6%), we claim that ***the larger MSE_D than MSE_S in P17 does not invalidate the previous conclusion, i.e., DISCOR consistently has better and more stable prediction accuracy than G-O, S-Shaped model and the three linear combinational models.***

6. Conclusions

There are over 40 different reliability growth models published to capture the law of failure process of software during testing phases, and to make the prediction of possible failures in the future. The failure process of real software during testing phases is very complicated because of the effects by human factors, bug distribution factors, test case factors, deployment factors, and the others. The effectiveness of reliability growth model usually varies in different projects. One reason is that single model can just capture partial knowledge of a complicated failure process by giving some necessary assumptions. Therefore, this motivates us to propose the combinational model, DISCOR, at the level of failure distribution. DISCOR model maintains the properties of NHPP so that we can do mathematical reliability analyses (e.g. failure rate analysis) based on DISCOR.

We use the maximum likelihood estimation and the popularly used likelihood function in this paper to estimate model parameters. To evaluate the effectiveness of DISCOR, we collect 21 public available data sets. After the exponential and gamma distribution tests, 12 data sets that have the *p-values* larger than 0.05 are selected; however, two of them cannot have feasible parameter estimation by MLE,. Therefore, we use the ten data sets to conduct the experiment and to evaluate DISCOR model. We compare DISCOR with its ancestors, G-O model, S-Shaped model, and the three linear combinational models, including ELC, MLC and ULC. We select eight sample points in each data set to calculate the mean squared error. The results show that the *MSEs* of DISCOR in the ten projects are stably lower than the other models, and we can observe

obvious fluctuations in the *MSE* measures of all the ten projects by the other models. Based on the measure of relative error reduced, we conclude that ***DISCOR reduces a significant percentage of MSE against the other models, and maintains the most stable MSE changes pattern in all the projects***.

Though the *MSE* of DISCOR model is not sensitive to the selection of the initial value of *N*, we still suggest selecting the initial value from the number of failures observed in a given data set. According to the experiment results, DISCOR model tends to have larger estimation of *N* than G-O and S-Shaped model. In particular, when a failure data set presents a short and thick tail in its density plot, DISCOR tends to produce the *N* estimation even several times larger than *NoF*. To our best knowledge, DISCOR is the first combinational model that tries to integrate existing reliability growth models at the level of failure distribution. The experiment results strongly supports that DISCOR outperforms the three linear combinational models, including ELC, MLC and ULC significantly. In the future, we will try to extend DISCOR to integrate other reliability growth models in the infinite category.

Since MLE does not guarantee to have feasible solutions in some cases, such as the two data sets P14c and PSS1A, we will study other parameter estimation methods, such as search-based method, to facilitate the application of DISCOR model better in the future.

Acknowledgements

This research is supported by the Technology Foundation Program (JSZL2014601B008) of the National Defense Technology Industry Ministry.

References

1. A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, “Evaluation of Competing Software Reliability Predictions,” *IEEE Transactions on Software Engineering*, vol. 12, no. 1, pp. 950-967, December 1986
2. S. Brocklehurst, P. Y. Chan, and B. Littlewood, “Recalibrating Software Reliability Model,” *IEEE Transactions on Software Engineering*, vol. 16, no. 4, pp. 458-470, April 1990
3. L. S. Dharmasena, P. Zeephongsekul, and C. L. Jayasinghe, “Software Reliability Growth Models based on Local Polynomial Modeling with Kernel Smoothing,” *International Symposium on Software Reliability Engineering (ISSRE). IEEE*, 2011
4. W. H. Farr, “A Survey of Software Reliability Modeling and Estimation,” *Technical Report NSWC TR 82-171, Naval Surface Weapons Center*, pp. 4-88, 1983
5. A. L. Goel, and K. Okumoto, “Time-dependent Error-detection Rate Model for Software Reliability and other Performance Measures,” *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206-211, 1979
6. J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, R. L. Tatham, Multivariate data analysis, *Pearson Prentice Hall*, Upper Saddle River, vol. 6, 2006
7. IEEE Std 1633-2008, IEEE Recommended Practice on Software Reliability. *New York: IEEE Reliabiliy Society*, 2008
8. T. Kim, K. Lee, and J. Baik, “An Effective Approach to Estimating the Parameters of Software Reliability Growth Models Using a Real-valued Genetic Algorithm,” *Journal of Systems & Software*, 102.C:134-144, 2015
9. B. Kitchenham, L. Madeyski, D. Budgen, K. Jacky, B. Pearl, C. Stuart, G. Shirley, and A. Pohthong, "Robust Statistical Methods for Empirical Software Engineering." *Empirical Software Engineering*, pp. 1-52, 2016
10. B. Littlewood, J. L. Verrall, “A Bayesian Reliability Growth Model for Computer Software,” *Journal of the Royal Statistical Society*, vol. 1, no. 22, pp.322-346, 1973
11. M. R. Lyu, “Measuring Reliability of Embedded Software: An Empirical Study with JPL Project Data,” pp.493-500, Feb 1991
12. M. R. Lyu, Handbook of Software Reliability Engineering, *McGraw-Hill Companies*, 1996
13. M. R. Lyu, and A. Nikora, “A Heuristic Approach for Software Reliability Prediction: The Equally-Weighted Linear Combination Model,” *International Symposium on Software Reliability*, pp. 172-181, June 1991
14. M. R. Lyu, and A. Nikora, “Software Reliability Measurements through Combination Models: Approaches, Results, and A CASE Tool,” *International Computer Software & Applications Conference*, pp. 577-584, 1991
15. R. E. Mullen, “The Lognormal Distribution of Software Failure Rates: Application to Software Reliability Growth Modeling[J],” vol. 117, pp. 134-142, 1998
16. J. D. Musa, DACS Software Reliability Dataset, *Data & Analysis Center for Software*, January 1980: http://www.dacs.dtic.mil/databases/sled/swrel.shtml
17. J. D. Musa, A. Iannino, and K. Okumoto, Software Reliability-Measurement, Prediction, Application, 1987
18. J. D. Musa, and K. Okumoto, “A Logarithmic Poisson Execution Time Model for Software Reliability Measurement,” *International Conference on Software Engineering*, pp. 230-238, Mar.1984
19. J. D. Musa, and K. Okumoto, “Software Reliability Models: Concepts, Classification, Comparisons, and Practics,” *Springer Berlin Heidelberg*, 1983
20. H. Okamura, Y. Etani, and T. Dohi, “A Multi-factor Software Reliability Model based on Logistic Regression,” *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on. IEEE*, 2010

21. J. Park, and J. Baik, "Improving Software Reliability Prediction through Multi-criteria based Dynamic Model Selection and Combination," *Journal of Systems and Software*, vol. 101, pp. 236-244, 2015
22. S. Ramasamy, and G. Govindasamy, "A Software Reliability Growth Model Addressing Learning," *Journal of Applied Statistics*, vol. 35, no. 10, pp. 1151-1168, 2008
23. N. F. Schneidewind, "Analysis of Error Processes in Computer Software," *Acm Sigplan Notices*, vol. 10, no. 6, pp. 337-346, 1975
24. K. Sharma, C. K. Nagpal, R. K. Garg, "Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach," *IEEE Transaction on Reliability*, vol. 59, no. 2, pp.266-276, 2010
25. Y.S. Su, C.Y. Huang, "Neural-network-based Approaches for Software Reliability Estimation Using Dynamic Weighted Combinational Models," *Journal of Systems & Software*, vol. 80, no. 4, pp.606-615, 2007
26. J. Wu, Shaukat Ali, T. Yue, J. Tian, and C. Liu, "Assessing the Quality of Industrial Avionics Software: an Extensive Empirical Evaluation," *Empirical Software Engineering*, pp. 1-50, 2016
27. S. Yamada, M. Ohba, and S. Osaki, "S-Shaped Reliability Growth Modeling for Software Error Detection," *IEEE Transactions on Reliability*, vol. 32, no. 5, pp. 475-484, 1983
28. J. Zhao, H. W. Liu, and C. Gang. "Software Reliability Growth Model Considering Testing Profile and Operation Profile," *Computer Software and Applications Conference. COMPSAC 2005. 29th Annual International. Vol. 1. IEEE*, 2005