

와이블 분포 특성에 근거한 소프트웨어 최적 방출시기에 관한 비교 연구

김희철¹, 박형근^{2*}

¹남서울대학교 산업경영공학과, ²남서울대학교 전자공학과

The Comparative Study of Software Optimal Release Time Based on Weibull Distribution Property

Kim, Hee Cheul¹ and Park, Hyoung Keun^{2*}

¹Department of Industrial Management Engineering, Namseoul University

²Department of Electronic Engineering, Namseoul University

요약 본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구하였다. 인도시기에 관한 모형은 무한 고장수에 의존하는 비동질적인 포아송 과정을 적용하였다. 이러한 포아송 과정은 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생할 가능성을 반영하는 모형이다. 고장발생 수명분포는 여러 분포들을 적합시키는데 효율적인 특성을 가진 와이블분포를 이용하였다. 따라서 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 방출시간이 최적 소프트웨어 방출 정책이 된다. 본 논문의 수치적인 예에서는 고장 간격 시간 자료를 적용하고 모수추정 방법은 최우추정법과 추세분석을 통하여 자료의 효율성을 입증한 후 최적 방출시기를 추정하였다.

Abstract In this paper, we were researched decision problem called an optimal release policies after testing a software system in development phase and transferring it to the user. The applied model of release time exploited infinite failure non-homogeneous Poisson process This infinite failure non-homogeneous Poisson process is a model which reflects the possibility of introducing new faults when correcting or modifying the software. The failure life-cycle distribution used the Weibull distribution which has the efficient various property which has the place efficient quality. Thus, optimal software release policies which minimize a total average software cost of development and maintenance under the constraint of satisfying a software reliability requirement becomes an optimal release policies. In a numerical example, after trend test applied and estimated the parameters using maximum likelihood estimation of inter-failure time data, estimated software optimal release time.

Key Words : Software Release Policies, Record Value Statistics, Weibull Distribution Property

1. 서론

소프트웨어 신뢰성은 컴퓨터 시스템에 대한 적응과 이에 대한 연구 분야에서 중요한 역할을 담당해 오고 있다. 소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 소프트웨어 신뢰도는 현대의 소프트웨어 생산품 개발에서 중

요한 분야 가운데 하나이다.

소프트웨어 신뢰성에 관한 연구 활동은 지난 30년 전 부터 행해져 오고 있고 많은 신뢰도 성장 모형들이 소프트웨어에 남아 있는 고장들의 수와 소프트웨어 신뢰도의 추정에 관한 문제들을 제안해 왔다.

일반적으로 소프트웨어 개발과정은 설계단계, 디자인, 코딩 그리고 테스트 단계로 거친다. 이러한 과정을 거친

*교신저자 : 박형근(phk315@nsu.ac.kr)

접수일 09년 04월 14일 수정일 (1차 09년 07월 02일, 2차 09년 07월 27일, 09년 08월 10일) 게재확정일 09년 08월 19일

후 소프트웨어 제품을 방출하게 되는데 방출이후에 발견되지 않은 고장들이 나타난다면 이것들에 대한 보전 비용(Maintenance cost)은 크게 증가 할 것이다. 결국, 소프트웨어 시스템 시험을 끝내고 그것을 사용자에게 넘기는 시기 결정은 매우 중요한 사항이 된다. 이러한 소프트웨어 방출시간에 대한 연구들은 대부분 유한 고장 NHPP(Non-Homogeneous Poisson Process)모형을 사용하였다[1-4]. 이러한 유한(Finite)고장 NHPP모형은 소프트웨어가 유한개의 고장이 있고 고장 제거 단계에서는 새로운 고장이 발생하지 않는다는 가정을 한 모형이다. 그러나 실제 고장 제거 단계에서도 새로운 고장이 발생할 수 있다. 따라서 본 연구에서는 무한(Infinte) 고장 NHPP 모형을 이용하여 최적 방출시기에 대한 문제를 제안하고자 한다.

이 분야에서는 Musa-Okumoto의 대수 포아송 실행시간 모형[5-7]과 로그-파우어 모형[7,8]을 이용한 방출 문제에 대한 문제 들이 이미 연구되었고 최근에도 이와 관련된 문제에 대한 연구는 Yang 과 Xie(2000) 와 Huang(2005)에 의해 보증기간 할인율 및 보증기간 이후의 사후대책 등에 대하여 연구되고 있다[9,10]. 본 연구에서는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생할 가능성이 있는 무한고장수를 가진 여러 분포들을 적합시키는데 효율적인 특성을 가진 와이블분포를 이용한 최적 방출시기에 관한 문제를 다루었다.

본 논문의 2절에서는 관련 연구로서 무한고장과 기록값 통계량, 로그 포아송 모수 모형 및 로그 파우어 모형 모수 추정에 대하여 약속하였고 3절에서는 요구 신뢰도와 비용 최소화를 고려한 방출시간에 대하여 서술하고 4장에는 제안된 와이블 특성 모형에 대하여 서술하고 5장과 6장에서는 각각 수치적인 예와 그 결론을 나열 하였다.

2. 관련연구

2.1 무한고장 NHPP와 기록값 통계량

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[2-4].

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (2.1)$$

따라서 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률 밀도 함수(Probability density function; Pdf)로 알려져 있다.

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty \quad (2.2)$$

이처럼 시간 관련 모형(Time domain models)들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 모형들은 고장 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수 $m(t)$ 도 역시 다르게 나타나고 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[12, 13]. 유한 고장 NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(Faults)의 기대값이 유한 값 ($\lim_{t \rightarrow \infty} m(t) = \theta < \infty$)을 가지고 반면에 무한 고장 NHPP 모형들은 무한 값을 가진다고 가정된다[11].

일반적으로 일반화된 순서 통계량(Generalized Order Statistics; GOS) 모형을 이용한 모형은 소프트웨어 테스트 시점에서 미지의 N 개의 결함들을 가지고 있고 이 N 개의 결함들로부터 임의의 확률 밀도 함수(Pdf)에 따라 발생된 n 개의 순서 통계량이 고장 시점이 된다. 이 모형은 각 수리 시점에서 새로운 결함이 발생하지 않는다는 가정을 한다. 그러나 실제 상황에서는 수리 시점에서도 고장이 발생할 수도 있다. 이러한 상황을 추가하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 사용하는 RVS(Record Value Statistics)모형을 사용할 수 있다고 하였고 이 RVS 모형과 NHPP 모형에 관해서 평균값함수는 다음과 같이 된다고 하였다[11-13].

$$m(t) = -\ln(1-F(t)) \quad (2.3)$$

즉, $e^{-m(t)} = 1-F(t)$.

따라서 (2.1)식과 (2.3)식과 연관시키고 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면 다음과 같은 관계식에 의해 NHPP의 강도함수는 $F(t)$ 의 위험함수($h(t)$)가 된다.

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (2.4)$$

결국 무한 고장 NHPP 모형의 평균값 함수와 고장 강도 함수는 각각 다음과 같이 유도된다[11,15].

$$m(t) = -\ln(1-F(t)) \quad (2.5)$$

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (2.6)$$

시간 $(0, t]$ 까지 조사하기 위한 시간 절단(Time truncated) 모형은 n 번째 까지 고장 시점 자료를

$$x_k = \sum_{i=1}^k t_k \quad (k = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (2.7)$$

이라고 하고 θ 을 모수공간이라고 하면 우도 함수는 다음과 같이 알려져 있다[11,13].

$$L_{NHPP_{NEV}}(\theta | D_t) = \prod_{i=1}^n \left(\frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n))$$

$$= \left(\prod_{i=1}^n h(x_i) \right) (1-F(x_n)) \quad (2.8)$$

2.2 기존모형

2.2.1 로그 포아송 실행시간모형

로그 포아송 실행시간(Log Poission execution time)모형[5,7,15]은 1984년에 Musa 와 Okumoto에 의해서 소개된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1) \quad (2.9)$$

$$\lambda(t) = \frac{\lambda_0}{\lambda_0 \theta t + 1} \quad (2.10)$$

한편, (2.3)식과(2.8)식을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\theta, \lambda_0 | \underline{x}) = \left(\prod_{i=1}^n \frac{\lambda_0}{\lambda_0 \theta x_i + 1} \right) \cdot e^{-\frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1)} \quad (2.11)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$.

모수 최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 다음과 같이 유도된다.

$$\ln L(\theta, \lambda_0 | \underline{x}) = n \ln \lambda_0 - \ln \sum_{i=1}^n (\lambda_0 \theta x_i + 1) - \frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1) \quad (2.12)$$

(2.12)식을 이용하여 최우추정치 $\hat{\theta}_{MLE}$ 와 $\hat{\lambda}_{0MLE}$ 은 다음과 같이 구할 수 있다고 하였다[5,6]

$$\hat{\theta}_{MLE} = \frac{1}{n} \ln(\hat{\phi} x_n + 1) \quad (2.13)$$

$$\hat{\lambda}_{0MLE} = \hat{\phi} / \hat{\theta}_{MLE} \quad (2.14)$$

단, $\phi (= \hat{\lambda}_{0MLE} \cdot \hat{\theta}_{MLE})$ 는 (2.15) 식의 근이 된다.

즉, 이 ϕ 근을 구하기 위해서는 수치 해석적 방법으로 다음과 같은 식을 이용하여 계산할 수 있다.

$$\frac{\partial \ln L(\phi | \underline{x})}{\partial \phi} = \frac{n}{\phi} - \sum_{i=1}^n \frac{x_i}{\phi x_i + 1} - \frac{n x_n}{(\phi x_n + 1) \ln(\phi x_n + 1)} = 0 \quad (2.15)$$

2.2.2 로그 파워어 (Log Power ; LP)모형

로그 파워어 (Log Power ; LP)모형[7,8]은 1999년에 Xie와 Homg에 의해서 발전된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = a \ln^b(1+t) \quad (2.16)$$

$$\lambda(t) = \frac{a b \ln^{b-1}(1+t)}{1+t} \quad (2.17)$$

한편, (2.3)식과 (2.8)식을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(a, b | \underline{x}) = \left(\prod_{i=1}^n \frac{a b \ln^{b-1}(1+x_i)}{1+x_i} \right) \cdot e^{-a \ln^b(1+x_n)}$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$.

(2.18)

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 다음과 같이 유도된다.

$$\ln L(a, b | \underline{x}) = n \ln a + n \ln b - (b-1) \ln \left(\sum_{i=1}^n \ln(1+x_i) \right) - \sum_{i=1}^n \ln(1+x_i) - a \ln^b(1+x_n) \quad (2.19)$$

(2.19)식에서 a 와 b 에 대하여 편미분 하여 다음과 같은 식을 만족하는 \hat{a}_{MLE} 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다.

$$\frac{\partial \ln L(a, b | \underline{x})}{\partial a} = \frac{n}{a} - \ln^b(1+x_n) = 0 \quad (2.20)$$

$$\frac{\partial \ln L(a, b | \underline{x})}{\partial b} = \frac{n}{b} - \ln \left(\sum_{i=1}^n \ln(1+x_i) \right) - a \ln^b(1+x_n) \ln(\ln(1+x_n)) = 0 \quad (2.21)$$

3. 요구 신뢰도 및 비용 최소화를 고려한 방출시간

NHPP 모형에서 테스트 시점 x_n (마지막 고장시점)에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(x_n, x_n + x]$ (단, x 는 임무시간(Mission time))동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도 $\hat{R}(x | x_n)$ 는 다음과 같이 됨이 알려져 있다[14, 15].

$$\begin{aligned} \hat{R}(x | x_n) &= \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) \\ &= \exp\{-[m(x+x_n)-m(x_n)]\} \end{aligned} \quad (3.1)$$

따라서 로그 포아송 실행시간모형에 대한 신뢰도는 평균값함수 (2.9)식과 $t = x_n$ 을 이용하면 다음과 같이 표현된다[5, 6].

$$\begin{aligned} R(x | t) &= \\ \exp\left(-\frac{1}{\theta} [\ln(\lambda_0\theta(x+t)+1) - \ln(\lambda_0\theta t+1)]\right) \end{aligned} \quad (3.2)$$

따라서 소프트웨어 방출시간 T_R 이 신뢰도 $R_0 = \hat{R}(x | t)$ 을 확보해야 한다면 다음 방정식을 만족해야 한다(방정식의 해).

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0\theta(x+T_R)+1) - \ln(\lambda_0\theta T_R+1)] \quad (3.3)$$

비용 최소화와 관련된 최적 방출시간은 신뢰도와 함께 비용모형에 의해서 결정된다. 소프트웨어 방출시간을 T 로 표현하고 $m(T)$ 와 $m(\infty)$ 을 각각 $(0, T]$ 와 $(0, \infty)$ 의 기간에 발견된 기대 고장수라고 표현하고 $C(T)$ 을 소프트웨어 라이프사이클(life cycle) 동안에 기대되는 소프트웨어 비용이라고 하면 $C(T)$ 는 다음과 같다[5, 6].

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (3.4)$$

위 식에서 c_1 는 테스트 동안에 하나의 고장을 수리하는 비용이고 c_2 가동 중에 하나의 고장을 수리하는 비용 ($c_2 > c_1$), 그리고 c_3 는 단위 시간당 테스트 비용을 나타낸다. 이와 관련하여 총비용의 최소화는 무한고장 평균값 함수를 가진 NHPP 모형에 대하여 발생 할 수 있다. 무한 수명에 대한 비용함수 $C(T)$ 식인 (3.1)과 (3.4)에서

$m(\infty)$ 은 직접 추정 할 수 없기 때문에 이 식을 사용하기 위해서는 소프트웨어 수명시간 인 T_{LC} 을 지정하여 분석한다[6]. 이러한 T_{LC} 는 소프트웨어마다 서로 다른 임의의 값이기 때문에 유한 고장 NHPP 모형이라고 할 수는 없다.

따라서 비용함수를 고려하여 소프트웨어의 모든 수명에서 총비용을 최소화함으로서 최적 테스트 시간을 결정할 수 있고 다음과 같은 식을 만족하면 비용함수 $C(T)$ 는 유일한 최소값을 가진다[5].

$$\frac{dC(T)}{dT} = 0, \quad \frac{d^2 C(T)}{d^2 T} > 0 \quad (3.5)$$

결국 소프트웨어 지정된 수명 T_{LC} 을 이용한 로그 포아송 실행시간모형 비용함수 $C(T)$ 는 다음과 같다.

$$\begin{aligned} C(T) &= c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \\ &= (c_1 - c_2) \frac{1}{\theta} \ln(\lambda_0\theta T + 1) \\ &\quad + \frac{c_2}{\theta} \ln(\lambda_0\theta T_{LC} + 1) + c_3 T \end{aligned} \quad (3.6)$$

T 에 관해서 비용함수 $C(T)$ 을 미분하면 최적방출시간 T_C 를 계산 할 수 있다[5, 6].

$$\frac{(c_1 - c_2)\lambda_0}{\lambda_0\theta T_C + 1} + c_3 = 0 \quad (3.7)$$

위 식에서 최적방출시간은 소프트웨어 지정 수명시간 인 T_{LC} 와 의존하지 않는다는 것을 알 수 있다. 이러한 사실은 무한 고장 평균값 함수를 가진 NHPP모형들이 새로운 결점들이 발생함으로서 몇 개의 고장이 야기 될 수 있는 점을 고려한 모형으로 적합시킬 수 있다[5]. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스트를 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 따라서 로그 포아송 실행시간모형을 사용한 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다 [6].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (3.8)$$

(3.8)식에서 T_R 과 T_C 는 다음 두 방정식에 의해서 계산 될 수 있다.

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0 \theta (x + T_R) + 1) - \ln(\lambda_0 \theta T_R + 1)] \quad (3.9)$$

$$\frac{(c_1 - c_2)\lambda_0}{\lambda_0 \theta T_C + 1} + c_3 = 0 \quad (3.10)$$

단, 신뢰도 $R_0 = \hat{R}(x | t)$

유사한 방법으로 로그 파우어 모형에서도 T_R 과 T_C 는 다음 두 방정식에 의해서 계산 될 수 있다.

$$\ln R_0 = -a(\ln^b(1 + (x + T_R)) - \ln^b(1 + T_R)) \quad (3.11)$$

$$(c_1 - c_2) \frac{a b \ln^{b-1}(1 + T_C)}{1 + T_C} + c_3 = 0 \quad (3.12)$$

4. 제안된 와이블 분포 특성을 고려한 모형

4.1 와이블 분포특성

와이블 분포는 감마 분포와 유사하게 적용성 때문에 신뢰도에서 중요한 역할을 한다. 신뢰도 함수에 포함된 모수로서 다른 분포들을 표현하거나 데이터를 적용하기 쉽게 변형 시킬 수 있다. 이러한 와이블 분포는 $\alpha(>0)$ 가 형상 모수, $\beta(>0)$ 는 척도 모수일 때, 확률 밀도 함수와 분포함수는 각각 다음과 같다.

$$f_{Weib}(t | \alpha, \beta) = \beta \alpha t^{\alpha-1} e^{-\beta t^\alpha} \quad (4.1)$$

$$F_{Weib}(t | \alpha, \beta) = 1 - e^{-\beta t^\alpha} \quad (4.2)$$

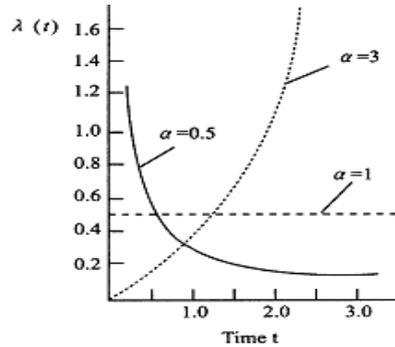
따라서 무한 고장 NHPP 모형의 평균값 함수 $m(t) = -\ln(1 - F(t)) = \beta t^\alpha$ 가 되고 강도함수는 다음과 같이 표현된다.

$$\lambda(t) = m'(t) = f(t)/(1 - F(t)) = \alpha \beta t^{\alpha-1} \quad (4.3)$$

이러한 와이블 분포는 많은 분포들에 적합시키는 데 사용될 수 있다고 하였고 다음과 같은 특성을 가지고 있다고 알려져 있다[11,15].

$0 < \alpha < 1$ 일 경우에는 (오류 수정 단계에서 발생하는) 감소하는 강도함수를 표현하고 $\alpha = 1$ 인 경우에는 (정상 운영 단계에서 발생하는) 일정한 강도함수를 나타내며 $\alpha > 1$ 인 경우에는 (마모 단계에서 발생하는) 증가하

는 강도함수를 표현한다. 이러한 패턴을 그림 1과 같이 나타낼 수 있다.



[그림 1] $\beta=0.5$ 인 경우 α 에 따른 강도함수 패턴

모수 최우추정법(MLE)을 이용하기 위한 로그 우도함수는 (2.8)식과 (4.3)식과 연관 하여 다음과 같이 유도된다.

$$\ln L(\alpha, \beta | \underline{x}) = n \ln \alpha + n \ln \beta + (\alpha - 1) \sum_{i=1}^n \ln x_i - \beta x_n^\alpha \quad (4.4)$$

(4.4)식을 이용하여 α 는 사전에 알고 있다고 가정하면 최우추정치 $\hat{\beta}_{MLE}$ 는 다음과 같이 구할 수 있다[5,6]

$$\hat{\beta}_{MLE} = \frac{n}{x_n^\alpha} \quad (4.5)$$

따라서 본 논문에서는 형상(shape)모수 α 가 0.5, 1.5, 2(Rayleigh분포)인 경우를 선택하여 분석 비교하고자 한다.

4.2 최적 방출시간 T_{OP}

신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 따라서 와이블 특성모형을 사용한 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다[6].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (4.6)$$

(4.6) 식에서 T_R 과 T_C 는 (3.1)식과 (3.6)식을 이용하면 다음 두 방정식에 의해서 계산 될 수 있다.

$$\ln R_0 = -\beta [(x + T_R)^\alpha - (T_R)^\alpha] \quad (4.7)$$

$$(c_1 - c_2) \alpha \beta T_C^{\alpha-1} + c_3 = 0 \quad (4.8)$$

단, 신뢰도 $R_0 = \hat{R}(x | T)$.

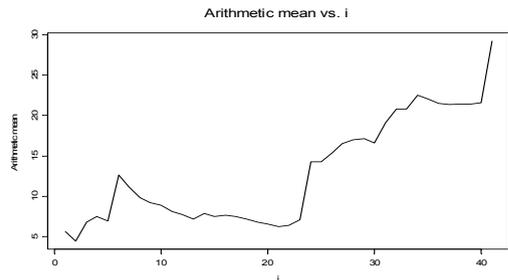
5. 수치적인 예

이 장에서 S27[16]가 인용한 고장 간격 시간 자료(Failure interval time data)를 가지고 와이블 특성 모형에 근거한 최적 방출시기를 분석하고자 한다. 이 자료는 1197.945 시간단위에 41번의 고장이 발생된 자료이며 표 1에 나열되어 있고 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다 [3,11].

[표 1] 고장 간격 자료

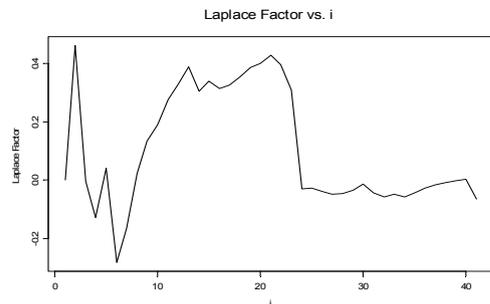
Failure number	Failure Time(hours)	Failure Interval (hours)
1	5.649	5.649
2	8.92	3.271
3	20.29	11.37
4	29.955	9.665
5	34.715	4.76
6	75.95	41.235
7	78.171	2.221
8	78.625	0.454
9	83.022	4.397
10	89.114	6.092
11	89.804	0.69
12	92.86	3.056
13	93.66	0.8
14	110.655	16.995
15	111.988	1.333
16	122.545	10.557
17	127.045	4.5
18	128.712	1.667
19	128.99	0.278
20	131.768	2.778
21	131.829	0.061
22	141.712	9.883
23	164.212	22.5
24	342.85	178.638
25	356.144	13.294
26	399.144	43
27	446.494	47.35
28	476.644	30.15
29	497.144	20.5
30	497.661	0.517
31	591.161	93.5
32	665.644	74.483
33	686.444	20.8
34	765.944	79.5
35	772.977	7.033
36	774.944	1.967
37	791.561	16.617
38	815.978	24.417
39	837.145	21.167
40	861.945	24.8
41	1197.945	336

추세 분석에는 산술평균 검정(Arithmetic mean test)과 라플라스 추세 검정(Laplace trend test)등이 있다. 이 검정을 실시한 결과 그림 2에서 산술평균 검정결과 고장시간이 증가함에 따라 산술 평균이 거의 증가 추세를 보이고 있으므로 신뢰성장(Reliability growth) 속성을 가지고 있고 그림 2의 라플라스 추세 검정의 결과도 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도와 소프트웨어 방출시기를 추정하는 것이 가능하다[16].



[그림 2] 산술평균검정

로그 포아송 실행시간모형과 로그 파워어 모형의 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산 방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10^{-2} 와 10 을, 허용 한계(Tolerance for width of interval)는 10^{-10} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다[17]. 그리고 와이블 특성 모형은 $\alpha = 0.5$ 와 $\alpha = 1.5, \alpha = 2$ 일 때 를 고정하여 모수 β 을 계산 한 값이 표 2에 요약되었다.



[그림 3] 라플라스 추세 검정

[표 2] 각 모형의 모수 추정값

Model	MLE	
Log Poission execution time	$\hat{\theta} = 0.020031$	$\hat{\lambda}_0 = 0.053067$
Log Power	$\hat{a} = 16.0583379$	$\hat{b} = 0.478584$
Weibull	$\alpha = 0.5$ (fixed)	$\hat{\beta} = 1.184583$
	$\alpha = 1.5$ (fixed)	$\hat{\beta} = 0.000989$
	$\alpha = 2.0$ (fixed)	$\hat{\beta} = 8.25 \times 10^{-7}$

[표 3] 최적 방출시간($R_0 = 0.95$)

Model	추정시간	최적방출 시간(T_{OP})
Log Poission execution time	$\hat{T}_R = 518.419706$	556.922319
	$\hat{T}_C = 556.922319$	
Log Power	$\hat{T}_R = 136.7676$	136.7676
	$\hat{T}_C = 102.5704$	
Weibull	$\alpha = 0.5$ $\hat{T}_R =$ 수렴안함 $\hat{T}_C = 4.215299$	추정 할 수 없음
	$\alpha = 1.5$ $\hat{T}_R = 12.67358$ $\hat{T}_C = 106.5092$	
	$\alpha = 2.0$ $\hat{T}_R = 247.8345$ $\hat{T}_C = 20202.02$	

표 3에서는 $c_1=5(\$)$, $c_2=20(\$)$ 그리고 $c_3=0.5(\$)$ 라고 가정하고 시스템 수명시간은 2000시간이고 임무시간을 x 을 1.5이고 R_0 을 0.95(95%)를 투입하여 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10와 1000을, 허용 한계(Tolerance for width of interval)는 10^{-10} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 추정을 수행하였다. 각 모형에 대한 추정시간의 결과와 최적방출시간은 표 3에 요약되었다.

이 표에서 최적방출시간은 $\alpha = 0.5$ 인 경우는 근이 수렴되지 않아서 방출시간을 추정할 수 없었지만 $\alpha = 1.5$ 인 경우는 기존의 로그 포아송 모형과 로그 파워어 모형에 비해 제안된 와이블 모형이 최적방출시간이 효율적임을 보여주고 있다. 그러나 $\alpha = 2.0$ 인 경우에는 비효율적으로 나타남을 볼 수 있었다. 따라서 와이블 분포 모형에서 형상모수가 작은 경우에는 와이블 모형도 이 분야에 새로운 모형으로 선택 할 수 있음을 보여주고 있다.

6. 결론

본 연구는 많은 분포들에 적합 시키는데 사용될 수 있는 와이블 분포를 적용한 무한고장 NHPP 모형을 이용하여 최적 방출시기에 관한 문제를 알아보았다. 즉, 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피 할 수 없는 상황이 현실이다. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스트를 계속해야 한다.

따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 본 연구에서는 기존의 로그 포아송 모형과 로그 파워어 모형에 비해 제안된 와이블 모형이 최적방출시간이 효율적임을 보여주고 있기 때문에 와이블 분포 모형도 이 분야에서 적용 가능한 모형이 될 수 있음을 확인하였다. 경우에 따라서는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화지수분포 등 업데이트된 분포에 대한 방출 시기 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 방출최적시기를 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료 된다.

참고문헌

- [1] Lawless, J. F, "Statistical Models and Methods for Lifetime Data", John Wiley & Sons, New York, 1981.
- [2] 김희철, "일반화감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한 연구", 한국컴퓨터정보학회논문지, 10권 6호, pp. 27-35, 2005.
- [3] S. S. Gokhale, K. S. Trivedi, "A time/structure based software reliability model", Annals of Software Engineering, pp.85-121, 1999.
- [4] 김희철, "지수화 지수 분포에 의존한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구", 한국컴퓨터정보학회논문지, 제11권 5호, pp.9-18, 2006.
- [5] Musa, J. D and Okumoto, K., "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Proceeding the 7th International Conference on Software Engineering, pp.230-238, 1984.
- [6] 김대경, "Musa-Okumoto의 대수 포아송 실행시간 모형에 근거한 비용-신뢰성 최적 정책", 품질경영학회지, 제 26권 3호, pp.141-149, 1988.
- [7] Almering, V. and Genuchten, M, V and Cloudt, G. and Sonnemans, "Using Software Reliability Growth Models in Practice", IEEE SOFTWARE, pp.82-88, 2007 .

- [8] Xie, M. and Homg, G. Y., "Software release time determination based on unbound NHPP model", Proceeding of the 24th International Conference on Computers and Industrial Engineering, pp.165-168, 1999.
- [9] Yang, B. and Xie. M., "A study of operational and testing reliability in software reliability analysis", RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol.70, pp.323-329, 2000.
- [10] Huang, C. Y., "Cost-Reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency, The journal of Systems and software, Vol.77, pp.139-155, 2005.
- [11] Kuo, L. and Yang, T. Y., "Bayesian Computation of Software Reliability", Journal of the American Statistical Association, Vol.91, pp.763-773, 1996.
- [12] 간광연, 이재곤, 김희철, "기록값 통계량 모형에 기초한 NHPP 소프트웨어 신뢰성에 관한 연구", 한국통신학회논문지, 제30권 12T호, pp.356-363, 2005.
- [13] 김희철, 신현철, 김경수, "기록값 통계량에 기초한 무한고장 NHPP 소프트웨어혼합 신뢰성장모형에 관한 연구", 정보보안논문지, 제7권 3호, pp. 51-60, 2007.
- [14] Pham, H. and Nordmann, L. and Zhang, X., "A General mperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate", IEEE Trans. on reliability, Vol.48, No.2, pp.169-175, 1999.
- [15] Musa, J. D, Iannino, A. and Okumoto, K., "Software Reliability: Measurement, Prediction, Application", McGraw Hill, New York, 1987 .
- [16] K. Kanoun , J. C. Laprie, "Handbook of Software Reliability Engineering", McGraw-Hill, New York, pp.401-437, 2003.
- [17] 김희철, "Log-Logistic 분포모형에 근거한 소프트웨어 최적방출시기에 관한 비교연구", 한국컴퓨터정보학회 논문지, 제13권 7호, pp.1-9, 2008.

김 희 철(Hee-Cheul Kim)

[정회원]



- 1992년 2월 : 동국대학교 대학원 통계학과 석사 (이학석사)
- 1998년 2월 : 동국대학교 대학원 통계학과 박사 (이학박사)
- 2000년 3월 ~ 2004년 2월 : 송호대학 정보산업계열 교수
- 2005년 3월 ~ 현재 : 서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어 신뢰성공학, 웹 프로그래밍, 전산통계

박 형 근(Hyoung-Keun Park)

[정회원]



- 1995년 2월 : 원광대학교 대학원 전자공학과 (공학석사)
- 2000년 2월 : 원광대학교 대학원 전자공학과 (공학박사)
- 1998년 5월 ~ 2001년 9월 : (주) 미디어서브기술연구소
- 2005년 3월 ~ 현재 : 남서울대학교 전자공학과 교수

<관심분야>

마이크로프로세서응용, 임베디드 S/W, 암호알고리즘