

강도함수 특성에 근거한 소프트웨어 최적 방출시기에 관한 비교 연구

김희철^{1*}, 박형근²

¹남서울대학교 산업경영공학과, ²남서울대학교 전자공학과

The Comparative Study of Software Optimal Release Time Based on Intensity Function property

Kim Hee Cheul^{1*} and Park Hyoung Keun²

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 연구되었다. 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. 강도함수는 여러 수명 분포들을 적합시키는데 효율적인 특성을 가진 콤펜즈, 파레토, 로그-로지스틱 패턴을 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 intensity function used Gompertz, Preto and Log-logstic pattern which has the efficient various property. 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, Infinite Non-Homogeneous Poisson Process, Intensity Function Property

1. 서론

소프트웨어 신뢰성에 관한 연구 활동은 지난 30년 전부터 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 단계를 거친다. 이러한 과정을 거친

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, 2]. 이러한 유한(Finite)고장 NHPP모형은 소프트

*교신저자 : 김희철(kim1458@nsu.ac.kr)

접수일 10년 02월 19일

수정일 10년 03월 26일

게재확정일 10년 04월 09일

웨어가 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[3]과 로그-파우어 모형[4]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 할인을 및 보증기간 이후의 사후대책 등에 대하여 연구되고 있다[5, 6]. 본 연구에서는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생할 가능성이 있는 무한고장수를 가진 여러 분포들을 적합시키는데 효율적인 특성을 가진 콤페르쯔, 파레토, 로그-로지스틱 강도함수를 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 콤페르쯔, 파레토, 로그-로지스틱 강도함수 특성 모형에 대하여 서술하고 5장과 6장에서는 각각 수치적인 예와 그 결론을 나열 하였다.

2. 관련연구

2.1 유한고장 및 무한고장 NHPP

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1].

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (1)$$

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

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

이러한 시간 관련 모형(Time domain models)들은 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수 $m(t)$ 도 역시 다르게 나타나는 NHPP에 의해서 확률 고장 과 정으로 표현되고 유한 고장 모형과 무한 고장 범주로 분

류한다[7]. 무한 고장 NHPP 모형들은 실제 상황에서는 수리 시점에서도 고장이 발생할 상황을 반영하기 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[7, 8].

$$m(t) = -\ln(1-F(t)) \quad (3)$$

따라서 식 (1)과 (3)을 연관시키고 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면 다음과 같은 관계식에 의해 NHPP의 강도함수는 $F(t)$ 의 위험함수($h(t)$)가 된다 [8].

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (4)$$

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

$$x_k = \sum_{i=1}^k t_k \quad (k=1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (5)$$

이라고 하고 θ 을 모수공간이라고 하면 우도 함수는 다음과 같이 알려져 있다[7, 9].

$$L_{NHPP_{NNV}}(\theta | D_t) = \prod_{i=1}^n \left(\frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n)) \quad (6)$$

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

2.2 기존모형

2.2.1 로그 포아송 실행시간모형

로그 포아송 실행시간(Log Poission execution time)모형[4, 10]은 1984년에 Musa 와 Okumoto에 의해서 소개된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1) \quad (7)$$

$$\lambda(t) = \frac{\lambda_0}{\lambda_0 \theta t + 1} \quad (8)$$

한편, 식 (3)과 (6)을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\theta, \lambda_0 | \underline{x}) = \left(\prod_{i=1}^n \frac{\lambda_0}{\lambda_0 \theta x_i + 1} \right) \cdot e^{-\frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1)} \quad (9)$$

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

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

$$\ln L(\theta, \lambda_0 | \underline{x}) = n \ln \lambda_0 - \ln \sum_{i=1}^n (\lambda_0 \theta x_i + 1) - \frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1) \quad (10)$$

식 (10)을 이용하여 최우추정치 $\hat{\theta}_{MLE}$ 와 $\hat{\lambda}_{0,MLE}$ 은 다음과 같이 구할 수 있다고 하였다.

$$\hat{\theta}_{MLE} = \frac{1}{n} \ln(\hat{\phi} x_n + 1) \quad (11)$$

$$\hat{\lambda}_{0,MLE} = \hat{\phi} / \hat{\theta}_{MLE} \quad (12)$$

단, $\phi (= \hat{\lambda}_{0,MLE} \cdot \hat{\theta}_{MLE})$ 는 (13) 식의 근이 된다.

즉, 이 ϕ 근을 구하기 위해서는 수치 해석적 방법으로 다음과 같은 식을 이용하여 계산할 수 있다[8].

$$\frac{\partial \ln L(\phi | \underline{x})}{\partial \phi} = \frac{n}{\phi} - \sum_{i=1}^n \frac{x_i}{\phi x_i + 1} - \frac{n x_n}{(\phi x_n + 1) \ln(\phi x_n + 1)} = 0 \quad (13)$$

2.2.2 로그 파워어 모형

로그 파워어 (Log power)모형[4, 11]은 1999년에 Xie 와 Homg에 의해서 발전된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = a \ln^b(1+t) \quad (14)$$

$$\lambda(t) = \frac{a b \ln^{b-1}(1+t)}{1+t} \quad (15)$$

한편, 식 (3)과 (6)을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(a, b | \underline{x}) = \left(\prod_{i=1}^n \frac{a b \ln^{b-1}(1+x_i)}{1+x_i} \right) \cdot e^{-a \ln^b(1+x_n)} \quad (16)$$

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

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는

다음과 같이 유도된다.

$$\ln L(a, b | \underline{x}) = n \ln a + n \ln b - (b-1) \ln \left(\sum_{i=1}^n \ln(1+x_i) \right) - \sum_{i=1}^n \ln(1+x_i) - a \ln^b(1+x_n) \quad (17)$$

식 (17)에서 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.

$$\frac{\partial \ln L(a, b | \underline{x})}{\partial a} = \frac{n}{a} - \ln^b(1+x_n) = 0 \quad (18)$$

$$\frac{\partial \ln L(a, b | \underline{x})}{\partial b} = \frac{n}{b} - \ln \left(\sum_{i=1}^n \ln(1+x_i) \right) - a \ln^b(1+x_n) \ln(\ln(1+x_n)) = 0 \quad (19)$$

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[1].

$$\hat{R}(x | x_n) = \exp \left(- \int_{x_n}^{x_n+x} \lambda(\tau) d\tau \right) = \exp \left[- \{m(x+x_n) - m(x_n)\} \right] \quad (20)$$

따라서 로그 포아송 실행시간모형에 대한 신뢰도는 평균값함수 식 (7)과 $t = x_n$ 을 이용하면 다음과 같이 표현 된다[3].

$$R(x | t) = \exp \left(- \frac{1}{\theta} [\ln(\lambda_0 \theta (x+t) + 1) - \ln(\lambda_0 \theta t + 1)] \right) \quad (21)$$

따라서 소프트웨어 방출시간 T_R 이 신뢰도 $R_0 = \hat{R}(x | t)$ 을 확보해야 한다면 다음 방정식을 만족해야 한다(방정식의 해)[8].

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0 \theta (x + T_R) + 1) - \ln(\lambda_0 \theta T_R + 1)] \quad (22)$$

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].

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (23)$$

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 (20)과 (23)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 최소화함으로써 최적 테스트 시간을 결정할 수 있고 다음과 같은 식을 만족하면 비용함수 $C(T)$ 는 유일한 최소값을 가진다[11].

$$\frac{dC(T)}{dT} = 0, \quad \frac{d^2 C(T)}{d^2 T} > 0 \quad (24)$$

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

$$\begin{aligned} C(T) &= c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (25) \\ &= (c_1 - c_2) \frac{1}{\theta} \ln(\lambda_0 \theta T + 1) \\ &\quad + \frac{c_2}{\theta} \ln(\lambda_0 \theta T_{LC} + 1) + c_3 T \end{aligned}$$

T 에 관해서 비용함수 $C(T)$ 을 미분하면 최적방출시간 T_C 를 계산 할 수 있다[8, 11].

$$\frac{(c_1 - c_2) \lambda_0}{\lambda_0 \theta T_C + 1} + c_3 = 0 \quad (26)$$

위 식에서 최적방출시간은 소프트웨어 지정 수명시간인 T_{LC} 와 의존하지 않는다는 것을 알 수 있다. 이러한 사실은 무한 고장 평균값 함수를 가진 NHPP모형들이 새로운 결점들이 발생함으로써 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 [11].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (27)$$

(27)식에서 T_R 과 T_C 는 다음 두 방정식에 의해서 계산될 수 있다.

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0 \theta (x + T_R) + 1) - \ln(\lambda_0 \theta T_R + 1)] \quad (28)$$

$$\frac{(c_1 - c_2) \lambda_0}{\lambda_0 \theta T_C + 1} + c_3 = 0 \quad (29)$$

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.

$$\ln R_0 = -a(\ln^b(1 + (x + T_R)) - \ln^b(1 + T_R)) \quad (30)$$

$$(c_1 - c_2) \frac{a b \ln^{b-1}(1 + T_C)}{1 + T_C} + c_3 = 0 \quad (31)$$

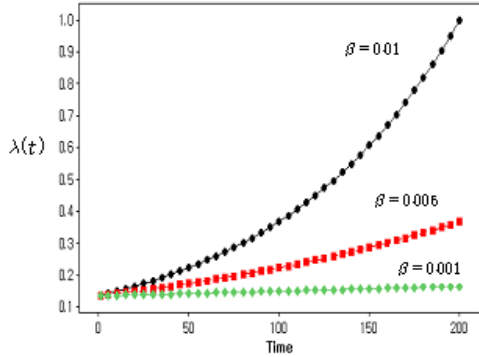
4. 제안된 강도함수 특성을 고려한 모형

4.1 강도함수가 콤펜트즈 패턴

콤펜트즈 분포(Gompertz)는 수명분포로 적용 할 수 있으며 다른 수명분포에 비해 실제적으로 효율적임을 보여 주었다[9]. 이러한 콤펜트즈 수명분포의 강도함수의 패턴은 그림 1에서 보여 주듯이 증가 추세를 가지며 다음과 같이 정의 된다.

$$\lambda(t) = \exp(\alpha + \beta t) \quad (32)$$

단, $0 < t < \infty, -\infty < \alpha < \infty, \beta > 0$.



[그림 1] $\alpha = -2$ 인 경우 β 에 따른 강도함수 패턴

따라서 평균값 함수는 식 (1) $m(t) = \int_0^t \lambda(s) ds$ 을 이용하면 다음과 같이 유도 할 수 있다.

$$m(t) = \frac{e^\alpha (e^{\beta t} - 1)}{\beta} \quad (33)$$

우도함수도 식 (6)을 이용하면 다음과 같이 표현된다.

$$L(\alpha, \beta | \underline{x}) = \exp \left(n\alpha + \beta \sum_{i=1}^n x_i - \frac{e^\alpha (e^{\beta x_n} - 1)}{\beta} \right) \quad (34)$$

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

(34)식에 의해서 α 와 β 에 관한 최우추정치 $\hat{\alpha}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 는 다음 식을 이용하면 추정 할 수 있다고 하였다[9].

$$\sum_{i=1}^n x_i + \frac{n}{\beta} - \frac{n x_n}{1 - e^{-\beta x_n}} = 0 \quad (35)$$

$$e^\alpha = \frac{n \beta}{e^{\beta x_n} - 1} \quad (36)$$

따라서 콤페르즈 특성을 사용한 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다[11].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (37)$$

(37) 식에서 T_R 과 T_C 는 식 (20)식과 (25)를 이용하면 다음 두 방정식에 의해서 계산 될 수 있다.

$$\ln R_0 = - \frac{e^\alpha}{\beta} [e^{\beta(T_R+x)} - e^{\beta T_R}] \quad (38)$$

$$T_C = \frac{\ln \left(\frac{-c_3}{(c_1 - c_2) e^\alpha} \right)}{\beta} \quad (39)$$

단, 신뢰도 $R_0 = \hat{R}(x | T)$, x 는 임무시간.

4.2 강도함수가 파레토 패턴

확률분포의 꼬리 부분이 정규분포보다 두꺼운 (Heavy-tailed) 즉, 다소 극단적인 값들이 발생할 확률을 무시할 수 없는 파레토(Pareto distribution) 분포의 확률 밀도 함수와 분포 함수는 각각 다음과 같다[7].

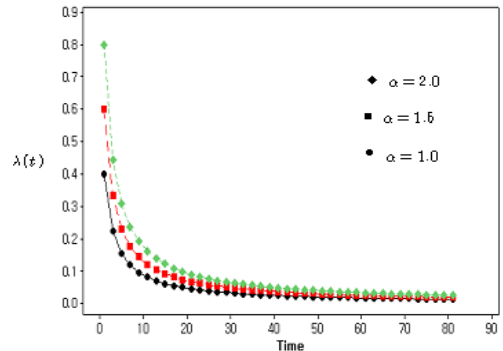
$$f_{Pareto}(t | \alpha, \beta) = \frac{\alpha \beta^\alpha}{(\beta+t)^{(1+\alpha)}} (\alpha, \beta > 0, t > 0) \quad (40)$$

$$F_{Pareto}(t | \alpha, \beta) = 1 - \left(\frac{\beta^\alpha}{(\beta+t)^\alpha} \right) \quad (41)$$

위 분포함수와 (3)식과 (4)을 이용하면 감소추세를 가진 강도함수와 평균값 함수는 다음과 같이 표현된다.

$$\lambda(t) = \frac{\alpha}{t+\beta} \quad (42)$$

$$m(t) = \alpha \ln \left(1 + \frac{t}{\beta} \right) \quad (43)$$



[그림 2] $\beta = 1.5$ 인 경우 α 에 따른 강도함수 패턴

식 (6)과 관련하여 무한 NHPP 고장 우도 함수는 다음과 같다[9].

$$L(\alpha, \beta | \underline{x}) = \left(\prod_{i=1}^n \frac{\alpha}{\beta + x_i} \right) \exp \left\{ -\alpha \ln \left(1 + \frac{x_n}{\beta} \right) \right\} \quad (44)$$

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

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 (44) 과 관련하여 다음과 같이 유도된다.

$$\ln L(\alpha, \beta | \underline{x}) = n \ln \alpha - \sum_{i=1}^n \ln(\beta + x_i) - \alpha \ln \left(1 + \frac{x_n}{\beta} \right) \quad (45)$$

(45)식에서 α 와 β 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\alpha}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다.

$$\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \alpha} = \frac{n}{\alpha} - \ln \left(1 + \frac{x_n}{\beta} \right) = 0 \quad (46)$$

$$\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \beta} = - \sum_{i=1}^n \frac{1}{(\beta + x_i)} - \alpha \left(\frac{1}{\beta + x_n} - \frac{1}{\beta} \right) = 0 \quad (47)$$

따라서 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다[11].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (48)$$

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

$$\ln R_0 = -\alpha \left\{ \ln \left(\frac{\beta + T_R + x}{\beta + T_R} \right) \right\} \quad (49)$$

$$T_C = \exp \left\{ \alpha \ln \beta + \frac{-c_3}{(c_1 - c_2) \alpha} \right\} - \beta \quad (50)$$

단, 신뢰도 $R_0 = \hat{R}(x | T)$, x 는 임무시간.

4.3 강도함수가 로그-로지스틱 패턴

로그-로지스틱(Log-Logistic)분포[1]는 강도함수(위험 함수)가 증가하다가 감소하는 속성을 가지고 있다. 이 분포의 확률밀도 함수와 분포함수는 다음과 같이 표현됨이 알려져 있다[12].

$$f(t | \tau, k) = (\lambda k (\tau t)^{k-1}) / [1 + (\tau t)^k]^2 \quad (51)$$

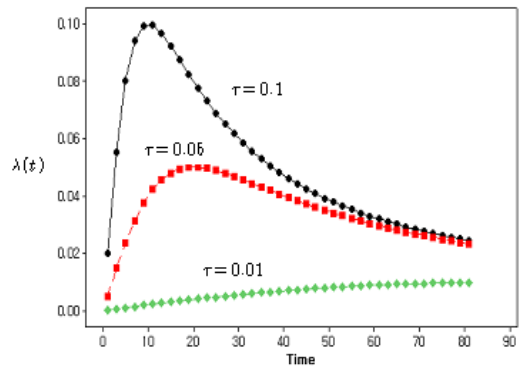
$$F(t | \tau, k) = (\tau t)^k / [1 + (\tau t)^k] \quad (52)$$

단, $t > 0$. 로그-로지스틱 모형을 (3)식과 (4)을 식을 이

용하여 무한 NHPP로 접근하면 평균값 함수와 강도 함수는 다음과 같이 표현 할 수 있다.

$$m(t | \tau, k) = \ln(1 + (\tau t)^k) \quad (53)$$

$$\lambda(t | \tau, k) = \frac{\tau k (\tau t)^{k-1}}{1 + (\tau t)^k} = \frac{\tau^k k t^{k-1}}{1 + (\tau t)^k} \quad (54)$$



[그림 3] $k=2.0$ 인 경우 τ 에 따른 강도함수 패턴

한편, 식 (6)을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\tau, k | \underline{x}) = \left(\prod_{i=1}^n \frac{\tau^k k x_i^{k-1}}{1 + (\tau x_i)^k} \right) \cdot [1 + (\tau x_n)^k]^{-1} \quad (55)$$

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

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 식 (55)와 관련하여 다음과 같이 유도된다.

$$\ln L(\tau, k | \underline{x}) = kn \ln \tau + n \ln k + (k-1) \sum_{i=1}^n \ln x_i - \sum_{i=1}^n \ln(1 + (\tau x_i)^k) - \ln[1 + (\tau x_n)^k] \quad (56)$$

(56)식에서 τ 와 r 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\tau}_{MLE}$ 와 \hat{k}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다[12]. 본 연구에서는 $k=2$ 일 경우를 고려 하였다.

$$\frac{\partial \ln L(\tau, k | \underline{x})}{\partial \tau} = \frac{kn}{\tau} - \sum_{i=1}^n \frac{x_i^k k \tau^{k-1}}{1 + (\tau x_i)^k} - \frac{k \tau^{k-1} x_n^k}{1 + (\tau x_n)^k} = 0 \quad (57)$$

$$\frac{\partial \ln L(\tau, k | x)}{\partial k} = n \ln \tau + \frac{n}{k} + \sum_{i=1}^n \ln x_i \quad (58)$$

$$- \sum_{i=1}^n \frac{\tau x_i (\ln \tau x_i)}{1 + (\tau x_i)^k} - \frac{\tau x_n (\ln \tau x_n)}{1 + (\tau x_n)^k} = 0$$

따라서 로그-로지스틱분포 모형을 사용한 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다[11].

$$T_{OP} = Max(T_C, T_R) \quad (59)$$

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

$$\ln R_0 = - \ln (1 + \{\tau(x + T_R)\}^k) + \ln (1 + (\tau T_R)^k) \quad (60)$$

$$\frac{(c_1 - c_2) k T_C^{k-1} \tau^k}{\ln(1 + (\tau T_C)^k)} + c_3 = 0 \quad (61)$$

단, 신뢰도 $R_0 = \hat{R}(x | T)$, x 는 임무시간.

5. 수치적인 예

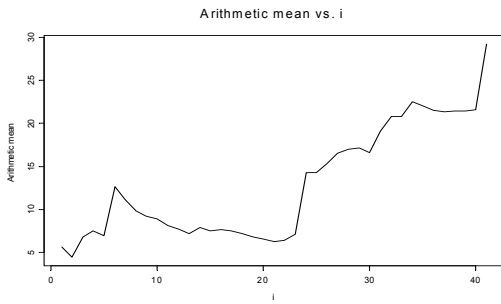
이 장에서 S27[13]가 인용한 고장 간격 시간 자료 (Failure interval time data)를 가지고 강도함수에 따른 최적 방출시기를 분석하고자 한다. 이 자료는 1197.945 시간단위에 41번의 고장이 발생한 자료이며 표 1에 나열되어 있고 제시하는 신뢰 모형을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[13].

추세 분석에는 산술평균 검정(Arithmetic mean test)과 라플라스 추세 검정(Laplace trend test)등이 있다. 이 검정을 실시한 결과 그림 4에서 산술평균 검정결과 고장시간이 증가함에 따라 산술 평균이 거의 증가 추세를 보이고 있으므로 신뢰성장(Reliability growth) 속성을 가지고 있고 그림 5의 라플라스 추세 검정의 결과도 라플라스 요인(Factor)이 -2와 2사이에 존재함으로서 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도와 소프트웨어 방출시기를 추정하는 것이 가능하다[13].

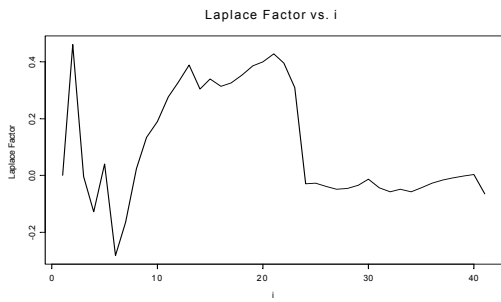
[표 1] 고장 간격 자료

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

각 모형의 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10^{-6} 와 20 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. 로그-로지스틱 모형은 $k=2$ 일 때를 고정하여 모수 τ 을 계산 한 값이 표 2에 요약되었다.



[그림 4] 산술평균검정



[그림 5] 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	$\hat{a} = 16.058338$	$\hat{b} = 0.478584$
Gompertz	$\alpha = -3.72358$	$\hat{\beta} = 0.000552$
Pareto	$\alpha = 6.345853$	$\hat{\beta} = 1.875762$
Log-Logistic	$k = 2.0$ (fixed)	$\hat{\tau} = 0.000069$

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.

[표 3] 최적 방출시간 T_{OP} ($R_0 = 0.95$)

Model	추정시간	T_{OP}
Log Poission execution time	$\hat{T}_R = 518.4197$ $\hat{T}_C = 556.9223$	556.9223
Log Power	$\hat{T}_R = 136.7676$ $\hat{T}_C = 102.5704$	136.7676
Gompertz	$\hat{T}_R = 629.5485$ $\hat{T}_C = 586.4359$	629.5485
Pareto	$\hat{T}_R = 182.9508$ $\hat{T}_C = 52.55281$	182.9508
Log-Logistic	$k = 2.0$ (Fixed) $\hat{T}_R = 99.4723$ $\hat{T}_C = 60.0005$	99.4723

이 표에서 제안된 콤페르즈 및 파레토 모형에 대한 최적방출시간은 강도함수가 증가하거나 감소하는 패턴을 따르기 때문에 기존의 로그 파워어 모형보다는 방출시간이 길어서 비효율적이지만 로그 포아송 모형보다는 효율적으로 나타나고 있다. 그러나 제안된 로그-로지스틱모형은 강도함수가 증가하다가 감소하는 패턴이기 때문에 우수한 모형으로 나타나고 있다. 결국 다른 자료를 적용시키면 다르게 나타날 수 있지만 강도함수가 변화하는(증가하다가 감소)하는 로그-로지스틱 모형이 우수한 모형이 될 수 있음을 알 수 있었다.

따라서 제안된 모형들도 이 분야에 새로운 모형으로 선택 할 수 있음을 보여주고 있다.

6. 결론

본 연구는 많은 분포들에 적합 시키는데 사용될 수 있는 강도함수의 패턴이 콤페르즈, 파레토 및 로그-로지스틱 패턴을 적용한 무한고장 NHPP 모형을 이용하여 최적 방출시간에 관한 문제를 알아보았다. 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 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 강도함수의 패턴이 증가, 감소 및 증가하다가 감소하는 패턴을 적용한 방출시기 모형을 시도한 결과 강도함수가 변화하는 모형이 효율적인 모형이 우수한 모형으로 나타남을 확인하였다. 강도함수에 영향을 주는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화지수분포 등 업데이트

트된 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] Gokhale, S. S. and Trivedi, K. S., "A time/structure based software reliability model", Annals of Software Engineering. 8, pp. 85-121, 1999.

[2] 김희철, "일반화감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한 연구", 한국컴퓨터정보학회논문지, 10권 6호, pp. 27-5, 2005.

[3] 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..

[4] Almering, V. and Genuchten, M, V and Cloudt, G. and Sonnemans, P. J. M., "Using Software Reliability Growth Models in Practice". IEEE SOFTWARE. pp. 82-88, 2007.

[5] 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.

[6] 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.

[7] Kuo, L. and Yang, T. Y., "Bayesian Computation of Software Reliability", Journal of the American Statistical Association, Vol,91, pp,763-773, 1996.

[8] 김희철, 박형근, "와이블분포특성에 근거한 소프트웨어 최적방출시기에 관한 비교연구", 한국 산학기술학회논문지, 제10권8호, pp. 1903-1910, 2009.

[9] Lawless, J. F "Statistical Models and Methods for Lifetime Data" John Wiley & Sons, New York, 1981.

[10] Musa, J. D, Iannino, A. and Okumoto, K., "Software Reliability: Measurement, Prediction, Application". McGraw Hill, New York, 1987.

[11] 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.

[12] 김희철, "Log-Logistic 분포모형에 근거한 소프트웨어 최적방출시기에 관한 비교연구", 한국 컴퓨터정보학회 논문지, 제13권7호, pp. 1-9, 2008.

[13] K. Kanoun, J. C. Laprie., Handbook of Software Reliability Engineering, M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, p.401-437, 1996.

김희철(Hee-Cheul Kim)

[정회원]



- 1992년 2월 : 동국대학교 통계학과 석사
- 1998년 2월 : 동국대학교 통계학과 박사
- 2000년 3월 ~ 2004년 2월 : 송호대학 정보산업계열 교수
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어 신뢰성공학, 웹 프로그래밍, 전산통계, 패널 데이터모형.

박형근(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