디지털산업정보학회 논문지 제6권 제3호-2010년 9월

중첩 NHPP 모형에 근거한 소프트웨어 최적방출시기에 관한 연구*

김 희 철**

A Study of Software Optimal Release Time for Based on Superposition NHPP Model

Kim, Hee Cheul

〈Abstract〉

Decision problem called an optimal release policies, after testing a software system in development phase and transfer it to the user, is studied. The applied model of release time exploited infinite non-homogeneous Poisson process. This infinite 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 superposition which has various intensity, if the system is complicated. Thus, 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. Through this study, in terms of superposition model and simply model, the optimal time to using superposition model release the software developer to determine how much could count will help.

Key Words : Software Release Policies, NHPP, Superposition, Optimal Release

Ⅰ. 서론

소프트웨어 방출시간에 대한 연구들은 대부분 유한 고장 NHPP(Non-Homogeneous Poisson Process)모형을 사용하였다[1-2]. 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 발생 할 수 있다. 따라서 본 연구에서는 무한(Infinite) 고장 NHPP 모형을 이용하여 최적 방출시기에 대한 문제를 제안하고자 한다.

최근까지도 이와 관련된 연구는 Yang과 Xie[3], Huang[4] 등에 의해 보증기간을 포함한 방출시기 등에 대하여 연구되고 있다[5].

본 연구에서는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생될 가능성이 있는 무한 고장수를 가진 비동질적인 포아송 과정에 기초하고 시스

* 이 논문은 2010학년도 남서울대학교 학술연구비 지원에 의하여 연구되었음.

** 남서울대학교 산업경영공학과 교수 (교신저자)

템이 복잡해지면 하나의 강도함수에만 고장이 일어나지 않고 다양한 원인에 의해 중첩되어 발생 할 수 있는 중첩모형을 이용한 방출시기에 관한 문제를 다루었다.

NHPP 모형에서 평균값 함수 $m(t)$(Mean value function)와 강도 함수(Intensity function) $\lambda(t)$는 다음과 같은 관계로 표현할 수 있다[2, 6].

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

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

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

이처럼 시간 관련 모형(Time domain model) 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[7].

수리 시점에서도 고장이 발생할 수도 있는 상황을 반영하기 위하여 RVS(Record Value Statistics)모형을 사용하는 NHPP 모형의 평균값함수는 다음과 같이 알려져 있다[2, 6].

$$m(t) = -\ln(1-F(t)) \tag{3}$$

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

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

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

$$x_k = \sum_{i=1}^{k} t_k \,(k = 1,2,\cdots,n\,;\, 0 \le x_1 \le x_2 \le \cdots \le x_n) \tag{5}$$

이라고 하고 Θ을 모수공간이라고 하면 우도 함수는 다음과 같이 알려져 있다[1, 7].

$$\begin{aligned} L_{NHPP_{INFN}}(\Theta \mid \underline{x}) &= \prod_{i=1}^{n} \left(\frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n)) \\ &= \left(\prod_{i=1}^{n} h(x_i) \right) (1-F(x_n)) \end{aligned} \tag{6}$$

단, $1-F(T) = e^{-m(t)}$, $\underline{x} = (x_1, x_2, x_3, \cdots, x_n]$

소프트웨어 시스템이 복잡해지면 하나의 강도함수에만 고장이 일어나지 않고 다양한 원인에 의해 중첩되어 발생 할 수 있다. 이런 경우에 강도함수와 평균값 함수를 알고 있다면 모형화가 가능하다[8].

$\{N(t)\,, t \ge 0\}$을 시점 t 까지 탐지된 소프트웨어 누적고장수를 나타내는 계수과정(Counting process)이라고 하고 t_i 을 고장발생시간 간격이고 x_n 는 n번째까지 발견된 고장시점이라고 하면 $x_n = \sum_{i=1}^{n} t_i$으로 표현 가능하고 데이터 집합은 다음과 같이 구성된다.

$$D = \{x_1\, x_2 \cdots, x_n\} \tag{7}$$

$m_j(t)$를 시점 $(0, t]$ 사이에서 강도함수 $\lambda_j(t)$을 따르는 j 번째 요소로부터 결함이 발생되는 NHPP라고 표현하면 $m(t) = \sum_{i=1}^{J} m_j(t)$(즉, j 번째 요소에 의해 발생된 고장의 수 $m_j(t)\,(j = 1,2,\cdots,J)$은 서로 독립이라고 가정)이면 $(0, t]$의 구간에서 발생된 고장수는 다음과 같

은 강도함수와 평균값함수를 따르는 중첩 NHPP가 된다[9].

$$m(t) = m_1(t) + m_2(t) + \cdots + m_J(t), \quad (8)$$

$$\lambda(t) = \lambda_1(t) + \lambda_2(t) + \cdots + \lambda_J(t), \quad (9)$$

따라서 $x_1 x_2 \cdots, x_n$을 소프트웨어 고장발생시점을 순서대로 나타내는 순서통계량 확률변수라고 가정하면 (6)식에 의존한 $x_1 x_2 \cdots, x_n$의 우도함수(Likelihood function)는 다음과 같이 표현된다[10].

$$L(D) = \prod_{k=1}^{n} \lambda(x_k) \cdot e^{-m(x_n)} \quad (10)$$

위 식에서 j 번째 요소로부터 고장이 발생되는 중첩 NHPP에 대한 우도함수는 다음과 같다[9].

$$L_{NHPP}(D_{S_n}) = \left[\prod_{k=1}^{n} \left\{\sum_{i=1}^{J} \lambda_i(x_k)\right\}\right] \cdot \exp\left\{-\sum_{i=1}^{J} m_i(x_n)\right\} \quad (11)$$

논문에서는 $j = 2$ 인 경우를 적용하고자 한다. 이 우도함수는 다음과 같다.

$$L_{NHPP}(D) = \left[\prod_{k=1}^{n} \{\lambda_1(x_k) + \lambda_2(x_k)\}\right] \cdot \exp\{-(m_1(x_n) + m_2(x_n))\} \quad (12)$$

Ⅱ. 관련연구

콤페르쯔 분포(Gompertz)[11]는 수명분포로 적용할 수 있으며 다른 수명분포에 비해 실제적으로 효율적이고 강도함수의 패턴은 증가 추세를 가지며 다음과 같이 정의 된다.

$$\lambda(t) = \exp(\alpha + \beta t) \quad (13)$$

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

따라서 평균값 함수는 (1)식 을 이용하면 다음과 같이 유도할 수 있다.

$$m(t) = \frac{e^{\alpha}(e^{\beta t} - 1)}{\beta} \quad (14)$$

우도함수는 (6) 식을 이용하면 다음과 같이 표현된다.

$$L(\alpha, \beta | \underline{x}) = \exp\left(2n + \beta \sum_{i=1}^{n} x_i - \frac{e^{\alpha}(e^{\beta x_n} - 1)}{\beta}\right) \quad (15)$$

단, $\underline{x} = (x_1 \le x_2 \le x_3, \cdots \le x_n]$ (15)식에서 $\frac{\partial \ln L(2, \beta | \underline{x})}{\partial \beta} = 0$와 $\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \alpha} = 0$을 만족하는 최우추정치 $\hat{\beta}_{MLE}$, $\hat{\alpha}_{MLE}$을 추정할 수 있다. 즉 다음을 만족한다[6].

$$\sum_{i=1}^{n} x_i + \frac{n}{\beta} - \frac{n x_n}{1 - e^{-\beta x_n}} = 0 \quad (16)$$

$$e^{\alpha} = \frac{n\beta}{e^{\beta x_n} - 1} \quad (17)$$

2.1 제안된 콤페르쯔 중첩모형의 패턴

수명 분포의 모수에는 크게 형상모수(Shape parameter)와 척도모수(Scale parameter)로 구분 해 볼 수 있다. 형상모수는 왜도와 첨도 등과 같이 확률함수의 모양에 관련된 모수이고 척도모수는 척도에 관한 측도를 의미한다. 본 절에서는 형상모수에 따른 중첩모형에 대하여 서술한다.

2.1.1 형상모수 $\beta=1$과 $\beta=2$ 모형의 중첩

콤페르쯔 파레토 중첩모형의 우도함수는 (12)식을 이용하면 다음과 같이 표현 가능하다[9].

$$L_{NHPP}(D)=\left[\prod_{k=1}^{n}\{\lambda_1(x_k)+\lambda_2(x_k)\}\right] \cdot \exp\{-(m_1(x_n)+m_2(x_n))\} \quad (18)$$

단, $\lambda_1(t)=\exp(\alpha+t)$, $\lambda_2(t)=\exp(\alpha+2t)$

$m_1(t)=\frac{e^{\alpha}(e^{t}-1)}{1}$, $m_2(t)=\frac{e^{\alpha}(e^{2t}-1)}{2}$

(18)식에서 $\frac{\partial \ln L_{NHPP}(D)}{\partial \alpha}=0$ 을 만족하는 최우추정치 $\hat{\alpha}_{MLE}$을 추정할 수 있다. 즉 다음을 만족한다.

$$2n+e^{\alpha}(3-2e^{x_n}-e^{2x_n})=0 \quad (19)$$

2.1.2 형상모수 $\beta=2$과 $\beta=3$ 모형의 중첩

콤페르쯔 파레토 중첩모형의 우도함수는 (12)식을 이용하면 다음과 같이 표현 가능하다.

$$L_{NHPP}(D)=\left[\prod_{k=1}^{n}\{\lambda_1(x_k)+\lambda_2(x_k)\}\right] \cdot \exp\{-(m_1(x_n)+m_2(x_n))\} \quad (20)$$

단, $\lambda_1(t)=\exp(\alpha+2t)$, $\lambda_2(t)=\exp(\alpha+3t)$

$m_1(t)=\frac{e^{\alpha}(e^{2t}-1)}{2}$, $m_2(t)=\frac{e^{\alpha}(e^{3t}-1)}{3}$

(20)식에서 $\frac{\partial \ln L_{NHPP}(D)}{\partial \alpha}=0$ 을 만족하는 최우추정치 $\hat{\alpha}_{MLE}$을 추정할 수 있다. 즉 다음을 만족한다.

$$6n+e^{\alpha}(5-3e^{2x_n}-2e^{3x_n})=0 \quad (21)$$

2.1.3 형상모수 $\beta=1$과 $\beta=3$ 모형의 중첩

콤페르쯔 파레토 중첩모형의 우도함수는 (12)식을 이용하면 다음과 같이 표현 가능하다.

$$L_{NHPP}(D)=\left[\prod_{k=1}^{n}\{\lambda_1(x_k)+\lambda_2(x_k)\}\right] \cdot \exp\{-(m_1(x_n)+m_2(x_n))\} \quad (22)$$

단, $\lambda_1(t)=\exp(\alpha+t)$, $\lambda_2(t)=\exp(\alpha+3t)$

$m_1(t)=\frac{e^{\alpha}(e^{t}-1)}{1}$, $m_2(t)=\frac{e^{\alpha}(e^{3t}-1)}{3}$

(22)식에서 $\frac{\partial \ln L_{NHPP}(D)}{\partial \alpha}=0$ 을 만족하는 최우추정치 $\hat{\alpha}_{MLE}$을 추정할 수 있다. 즉 다음을 만족한다.

$$3n+e^{\alpha}(4-3e^{2x_n}-e^{3x_n})=0 \quad (23)$$

Ⅲ. 신뢰도 및 비용최소화를 고려한 방출시간

NHPP 모형에서 테스트 시점 x_n(마지막 고장시점)에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(x_n, x_n+x]$ (단, x는 임무시간(Mission time))동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도 $\hat{R}(x \mid x_n)$는 다음과 같이 됨이 알려져 있다[6, 11].

$$\hat{R}(x \mid x_n)=\exp\left(-\int_{x_n}^{x_n+x}\lambda(\tau)d\tau\right) = \exp[-\{m(x+x_n)-m(x_n)\}] \quad (24)$$

따라서 콤페르쯔 분포 모형의 평균값 함수 (15)식을 이용하면 신뢰도는 (25)식에서 $t = x_n$라고 치환하면 다음과 같이 표현된다.

$$R(x|t) = \exp\left[-\frac{e^{\alpha}}{\beta}\left\{ e^{\beta(T_R+x)} - e^{\beta T_R} \right\} \right] \quad (25)$$

따라서 소프트웨어 방출시간 T_R이 신뢰도 $R_0 = \hat{R}(x \mid T_R)$을 확보해야 한다면 다음 방정식을 만족해야 한다.

$$\ln R_0 = -\frac{e^{\alpha}}{\beta}\left[e^{\beta(T_R+x)} - e^{\beta T_R} \right] \quad (26)$$

단, 신뢰도 $R_0 = \hat{R}(x \mid T_R)$.

한편, 소프트웨어 방출시간을 T로 표현하고 $m(T)$와 $m(\infty)$을 각 각 $(0, T]$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 같이 표현된다[12].

$$C(T) = c_1 m(T) + c_2[m(\infty) - m(T)] + c_3 T \quad (27)$$

위 식에서 c_1는 테스팅 동안에 하나의 고장을 수리하는 비용이고 c_2 가동 중에 하나의 고장을 수리하는 비용 $(c_2 > c_1)$, 그리고 c_3는 단위 시간당 테스팅 비용을 나타낸다. 이와 관련하여 총비용의 최소화는 무한고장 평균값 함수를 가진 NHPP 모형에 대하여 발생 할 수 있다. 무한 수명에 대한 비용함수 $C(T)$식인 (24)식에서 $m(\infty)$은 직접 추정 할 수 없기 때문에 이 식을 사용하기 위해서는 소프트웨어 수명시간 인 T_{LC} 을 지정하여 분석한다[12]. 이러한 T_{LC}는 소프트웨어마다 서로 다른 임의의 값이기 때문에 유한 고장 NHPP 모형이라고 할 수 는 없다.

따라서 비용함수를 고려하여 소프트웨어의 모든 수명에서 총비용을 최소화함으로서 최적 테스팅 시간을 결정할 수 있고 다음과 같은 식을 만족하면 비용함수 $C(T)$는 유일한 최소값을 가진다[12-13].

$$\frac{\partial C(T)}{\partial T} = 0, \quad \frac{\partial^2 C(T)}{\partial^2 T} > 0 \quad (28)$$

결국 콤페르쯔 분포에 의한 최소비용 관련 방출시간은 (27)식을 연관하여 다음과 같이 유도 된다.

$$\begin{aligned} C(T) &= c_1 m(T) + c_2[m(\infty) - m(T)] + c_3 T \\ &= (c_1 - c_2)\left(\frac{e^{\alpha}(e^{\beta T} - 1)}{\beta} \right) + c_2\left(\frac{e^{\alpha}(e^{\beta T_{LC}} - 1}{\beta} \right) + c_3 T \end{aligned} \quad (29)$$

T 에 관해서 비용함수 $C(T)$을 미분하면 $\frac{\partial C(T)}{\partial T} = 0$을 만족하는 최적방출시간 T_C를 계산 할 수 있다.

$$T_C = \frac{\ln\left(\frac{-c_3}{(c_1 - c_2)\, e^{\alpha}} \right)}{\beta} \quad (30)$$

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 발생함으로서 몇 개의 고장이 야기 될 수 있는 점을 고려한 모형으로 적합시킬 수 있다[13].

실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스팅을 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 따라서 콤페르쯔 모형을 사용한 최적 방출시간 T_{OP} 는 T_R과 T_C에 대하여 다음을 만족한다[13].

$$T_{OP} = Max(T_C, T_R) \quad (31)$$

단,
$$\ln R_0 = -\frac{e^{\alpha}}{\beta}\left[e^{\beta(T_R+x)} - e^{\beta T_R}\right] \quad (32)$$

$$T_C = \frac{\ln\left(\frac{-c_3}{(c_1-c_2)\,e^{\alpha}}\right)}{\beta} \quad (33)$$

유사한 방법으로 중첩모형의 경우는 T_R과 T_C는 다음과 같이 계산 될 수 있다.

<표 1> 중첩 모형에 대한 T_R와 T_C

중첩모수	T_R 과 T_C
$\beta=1$ $\beta=2$	$\ln R_0 = \frac{1}{2}\left[2e^{T_R+\alpha}(1-e^{x}) - e^{2T_R+\alpha}(1-e^{2x})\right]$ $(c_1-c_2)\,e^{\alpha}\left(e^{T_C}+e^{2T_C}\right)+c_3=0$
$\beta=2$ $\beta=3$	$\ln R_0 = \frac{1}{6}\left[3e^{2T_R+\alpha}(1-e^{2x}) - 2e^{3T_R+\alpha}(1-e^{3x})\right]$ $(c_1-c_2)\,e^{\alpha}\left(e^{2T_C}+e^{3T_C}\right)+c_3=0$
$\beta=1$ $\beta=3$	$\ln R_0 = \frac{1}{3}\left[3e^{T_R+\alpha}(1-e^{x}) - e^{3T_R+\alpha}(1-e^{3x})\right]$ $(c_1-c_2)\,e^{\alpha}\left(e^{T_C}+e^{3T_C}\right)+c_3=0$

V. 수치적인 예

이 장에서는 고장 간격 시간 자료(Failure interval time data)[14]를 적용하고 콤페르쯔 분포에 근거한 형상모수의 중첩 모형에 대한 최적 방출시기를 분석하고자 한다.

이 자료는 1197.945 시간단위에 41번의 고장이 발생된 자료이며 <표 2>에 나열 되어 있고 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[14].

추세 분석에는 일반적으로 라플라스 추세 검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 <그림 1>에서 라플라스 추세 검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로서 신뢰성장

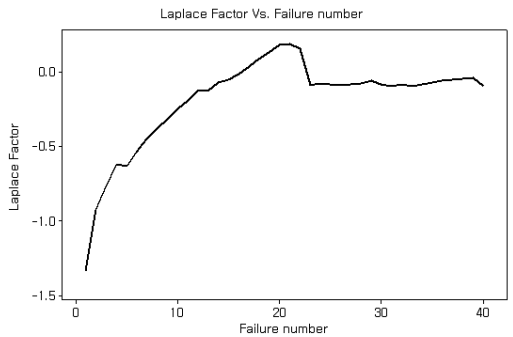
<표 2> 고장 자료

Failure number	Failure Time (hours)	Failure Interval (hours)	$Failure\ time \times 10^{-3}$
1	5.649	5.649	0.005649
2	8.92	3.271	0.00892
3	20.29	11.37	0.02029
4	29.955	9.665	0.029955
5	34.715	4.76	0.034715
6	75.95	41.235	0.07595
7	78.171	2.221	0.078171
8	78.625	0.454	0.078625
9	83.022	4.397	0.083022
10	89.114	6.092	0.089114
11	89.804	0.69	0.089804
12	92.86	3.056	0.09286
13	93.66	0.8	0.09366
14	110.655	16.995	0.110655
15	111.988	1.333	0.111988
16	122.545	10.557	0.122545
17	127.045	4.5	0.127045
18	128.712	1.667	0.128712
19	128.99	0.278	0.12899
20	131.768	2.778	0.131768
21	131.829	0.061	0.131829
22	141.712	9.883	0.141712
23	164.212	22.5	0.164212
24	342.85	178.638	0.34285
25	356.144	13.294	0.356144
26	399.144	43	0.399144
27	446.494	47.35	0.446494
28	476.644	30.15	0.476644
29	497.144	20.5	0.497144
30	497.661	0.517	0.497661
31	591.161	93.5	0.591161
32	665.644	74.483	0.665644
33	686.444	20.8	0.686444
34	765.944	79.5	0.765944
35	772.977	7.033	0.772977
36	774.944	1.967	0.774944
37	791.561	16.617	0.791561
38	815.978	24.417	0.815978
39	837.145	21.167	0.837145
40	861.945	24.8	0.861945
41	1197.945	336	1.197945

(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도와 소프트웨어 방출시기를 추정하는 것이 가능하다[10, 14].

<표 3>에서는 이 자료에 대한 기초통계량이 요약되어 있다. 이 표에서 왜도와 첨도가 각각 0.86과 -0.41로 나타나 분포의 꼬리가 오른쪽에 있고 대부분의 관측 값들이 왼쪽에 분포되어 있음을 의미하고 비교적 관측 값이 넓게 분포되는 형태를 보여주고 있다.

본 논문에서는 모수추정을 용이하게 하기 위하여 실제자료에서 수치 변환된 자료($failure\ times \times 10^{-3}$)를 이용 하였다. 각 모형의 모수 추정값인 최우추정치는 단순모형인 콤페르쯔 모형에 대해서는 형상모수 β를 1, 2, 3으로 고정하고 중첩모형에 대해서는 형상모수 β를 1과 2, 2와3, 1과 3인 중첩패턴을 이용하였다. 그 결과는 <표 4. >에 요약되었다.



<그림 1> 라플라스 추세 검정

<표 3> 고장 시간자료의 기술통계량

기술 통계량	
평균	0.338
중앙값	0.132
첨도	-0.41
왜도	0.86
범위	1.1923
관측수	41
신뢰 구간(95.0%)	(0.237, 0.439)

<표 4> 각 모형의 모수 추정 값

Model		MLE
Gompertz	$\beta=1$	$\alpha=2.875$
	$\beta=2$	$\alpha=2.106$
	$\beta=3$	$\alpha=1.246$
Superposition	$\beta=1\ \ \beta=2$	$\alpha=1.725$
	$\beta=2\ \ \beta=3$	$\alpha=1.067$
	$\beta=1\ \ \beta=3$	$\alpha=0.893$

<표 5> 각 모형의 최적 방출시간 계산($R_0=0.95$)

Model		추정시간	최적방출시간 (T_{OP})
Gompertz	$\beta=1$	$\widehat{T}_R=1.755$ $\widehat{T}_C=1.142$	1.755
	$\beta=2$	$\widehat{T}_R=1.262$ $\widehat{T}_C=1.218$	1.262
	$\beta=3$	$\widehat{T}_R=0.841$ $\widehat{T}_C=1.672$	1.672
Superposition	$\beta=1$ $\beta=2$	$\widehat{T}_R=1.569$ $\widehat{T}_C=0.292$	1.569
	$\beta=2$ $\beta=3$	$\widehat{T}_R=1.295$ $\widehat{T}_C=0.866$	1.295
	$\beta=1$ $\beta=3$	$\widehat{T}_R=1.272$ $\widehat{T}_C=0.999$	1.272

<표 5>에서는 c_1=0.001, c_2=0.01($) 그리고 c_3=0.5($)라고 가정하고 시스템 수명시간은 20시간이고 임무시간 x는 0.0005이고 R_0을 0.95(95%)를 투입하여 각 모형에 대한 추정시간의 결과와 최적방출시간이 요약되었다.

비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 0.1 와 5 을, 허용 한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 이 표에서는 단순모형인 콤페르쯔 모형에서는 형상모수가 $\beta=2$ 인 경우는 최적 방출시간이 다른

모형보다 길지 않아 효율적이므로 우수한 모형으로 나타나고 있다. 그러나 그 외에는 형상모수의 중첩모형이 효율적인 모형으로 나타나고 있으며 단순모형이든 중첩모형이든 형상모수 값이 클수록 신뢰측면 방출시간이 효율적이고 비용측면 방출시간은 비효율적으로 나타나고 있다. 결국 제안된 중첩모형은 기존의 단순 모형 비해 최적 방출시간이 효율적임을 보여주고 있다. 이 결과는 적합시킨 자료가 달라지면 그 결과는 달라질 수 있지만 형상모수의 중첩모형도 이 분야에서 우수한 모형이 될 수 있음을 확인 할 수 있었다.

Ⅵ. 결론

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 테스팅을 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다.

본 연구에서는 소프트웨어의 결함을 제거하거나 수정작업 중에도 새로운 결함이 발생될 가능성이 있는 무한고장수를 가진 비동질적인 포아송 과정에 기초하고 시스템이 복잡해지면 하나의 형상모수(강도함수)에만 고장이 일어나지 않고 다양한 원인에 의해 혼합되어 발생 할 수 있는 형상모수의 중첩모형을 이용한 최적방출시기에 관한 문제를 다루었다. 그 결과 단순모형 뿐만 아니라 형상모수의 중첩모형도 우수한 모형이 될 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 방출최적시기를 단순 보다는 중합 모형에 대한 사전 지식을 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료된다.

참고문헌

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

[2] 김희철, “지수화 지수 분포에 의존한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구,” 한국 컴퓨터 정보학회 논문지, 제11권 5호, 2006, pp. 9-18.

[3] Yang, B. and Xie. M, “A study of operational and testing reliability in software reliability analysis,” RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol. 70, 2000, pp. 323-329.

[4] 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, 2005, pp. 139-155

[5] 김희철, “극값 분포 특성을 가진 소프트웨어 신뢰성 보증 모형에 관한 연구,” 한국 통신학회 논문지, 34권6호, 2009, pp. 623-629.

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

[7] Kuo, L. and Yang, T. Y, “Bayesian Computation of Software Reliability,” Journal of the American Statistical Association, Vol. 91, 1996, pp. 763-77.

[8] 김희철, 신현철 “중첩커버리지 함수를 고려한 ENHPP 소프트웨어 신뢰성장 모형에 관한 연구,” 정보, 보안 논문지, 제7권3호, 2007, pp. 7-13.

[9] 김도훈, 남경현, “중첩 NHPP를 이용한 소프트웨어 신뢰도 평가 모형 연구,” 품질경영학회지, 제36권 1호, 2008, pp. 89-95.

[10] 김재욱, 김희철 “일반화 감마분포에 근거한 소프트웨어 최적방출시기에 관한 연구,” 디지털산업정보

학회논문지, 제6권 1호, 2010, pp. 55-67.

[11] 김희철, 박형근 “강도함수 특성에 근거한 소프트웨어 최적 방출시기에 관한 비교 연구,” 한국 산학기술 학회 논문지, 제11권4호, 2010, pp. 1239-1247.

[12] 이상식, 김희철 “혼합 와이블 NHPP 모형에 근거한 소프트웨어 최적방출시기에 관한 연구,” 디지털산업정보학회논문지, 제6권 2호, 2010, pp. 183-191.

[13] 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, 1996, pp. 165-168.

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

■ 저자소개 ■

김 희 철
Kim, Heel Cheul

2005년 3월~현재
남서울대학교 산업경영공학과 교수
1998년 2월 동국대학교 통계학과(이학박사)
1992년 2월 동국대학교 통계학과(이학석사)

관심분야 : 소프트웨어 신뢰성공학,
웹 프로그래밍
E-mail : kim1458@nsu.ac.kr

논문접수일 : 2010년 7월 22일
수 정 일 : 2010년 8월 27일
게재확정일 : 2010년 9월 3일