

# 기록값 통계량을 이용한 소프트웨어 최적 방출시기에 관한 비교 연구

김희철\*, 박형근\*\*

\*남서울대학교 산업경영공학과

\*\*남서울대학교 전자공학과

e-mail:phk315@nsu.ac.kr

## The Comparative Study of Software Optimal Release Time Based on Record Value Statistics

Hee-Cheul Kim\*, Hyoung-Keun Park\*\*

\*Dept of Industrial Management Eng., Namseoul University

\*\*Dept of Electronic Eng., Namseoul University

### 요 약

본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구되었다. 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생될 가능성이 있는 무한고장수를 가진 비동질적인 포아송 과정에 기초하고 수명분포는 신뢰도 측면에서 여러 분포들을 적합시키는데 효율적인 특성을 가진 와이블분포를 이용한 최적 방출시기에 관한 문제를 제시하여 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 최적 소프트웨어 방출 정책에 대하여 논의되었다.

### 1. 서론

소프트웨어 방출시간에 대한 연구들은 대부분 유한 고장 NHPP (Non-Homogeneous Poisson Process)모형을 사용하였다[1,2,3,4]. 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 발생 할 수 있다. 따라서 본 연구에서는 무한(infinite) 고장 NHPP 모형을 이용하여 최적 방출시기에 대한 문제를 제안하고자 한다. 이 분야에서는 Musa-Okumoto의 대수 포아송 실행 시간 모형[5,6,7] 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 연구되고 있다[9,10]. 본 연구에서는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생될 가능성이 있는 무한고장수를 가진 여러 분포들을 적합시키는데 효율적인 특성을 가진 와이블분포를 이용한 최적 방출시기에 관한 문제를 다루었다.

### 2. 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].

$$\hat{R}(x | x_n) = \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) = \exp\{-[m(x+x_n) - m(x_n)]\} \quad (2.1)$$

따라서 로그 포아송 실행시간모형에 대한 신뢰도는 평균값 함수와  $t=x_n$ 을 이용하면 다음과 같이 표현된다[5,6].

$$R(x | t) = \exp\left(-\frac{1}{\theta} [\ln(\lambda_0\theta(x+t)+1) - \ln(\lambda_0\theta t+1)]\right) \quad (2.2)$$

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

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0\theta(x+T_R)+1) - \ln(\lambda_0\theta T_R+1)] \quad (2.3)$$

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 같이 표현된다[5,6].

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (2.6)$$

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 (2.3)과 (2.6)에서  $m(\infty)$ 은 직접 추정 할 수 없기 때문에 이 식을 사용하기 위해서는 소프트웨어 수명시간 인  $T_{LC}$ 을 지정하여 분석한다 [6]. 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[5].

$$\frac{dC(T)}{dT} = 0, \quad \frac{d^2 C(T)}{dT^2} > 0 \quad (2.7)$$

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

$$\begin{aligned} C(T) &= c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (2.8) \\ &= (c_1 - c_2) \frac{1}{\theta} \ln(\lambda_0 \theta T + 1) \\ &\quad + \frac{c_2}{\theta} \ln(\lambda_0 \theta T_{LC} + 1) + c_3 T \end{aligned}$$

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

$$\frac{(c_1 - c_2) \lambda_0}{\lambda_0 \theta T_C + 1} + c_3 = 0 \quad (2.9)$$

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 발생함으로써 몇 개의 고장이 야기 될 수 있는 점을 고려한 모형으로 적합 시킬 수 있다[5]. 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].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (2.10)$$

(3.9) 식에서  $T_R$ 과  $T_C$ 는 다음 두 방정식에 의해서 계산된다.

$$\ln R_0 = -\frac{1}{\theta} [\ln(\lambda_0 \theta (x + T_R) + 1) - \ln(\lambda_0 \theta T_R + 1)] \quad (2.11)$$

$$\frac{(c_1 - c_2) \lambda_0}{\lambda_0 \theta T_C + 1} + c_3 = 0 \quad (2.12)$$

단, 신뢰도  $R_0 = \hat{R}(x | t)$

### 3. 제안된 와이블 분포 특성을 고려한 모형

#### 3.1 와이블 분포 특성

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.

$$f_{Wei}(t | \alpha, \beta) = \beta \alpha t^{\alpha-1} e^{-\beta t^\alpha} \quad (3.1)$$

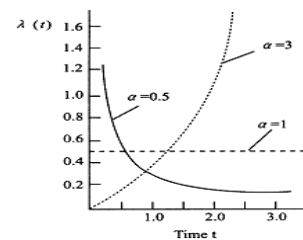
$$F_{Wei}(t | \alpha, \beta) = 1 - e^{-\beta t^\alpha} \quad (3.2)$$

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

$$\lambda(t) = m'(t) = f(t) / (1 - F(t)) = \alpha \beta t^{\alpha-1} \quad (3.3)$$

이러한 와이블 분포는 많은 분포들에 적합시키는데 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, 이러한 패턴을 [그림 1]과 같이 나타낼 수 있다.



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

모수 최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 (3.3)식과 연관 하여 다음과 같이 유도된다.

$$\ln L(\alpha, \beta | \underline{x}) = n \ln \alpha + n \ln \beta + (\alpha - 1) \sum_{i=1}^n \ln x_i - \beta x_n^\alpha \quad (3.4)$$

(3.4)식을 이용하여  $\alpha$ 는 사전에 알고 있다고 가정하면 최우 추정치  $\hat{\beta}_{MLE}$ 는 다음과 같이 구할 수 있다고 하였다[5,6]

$$\hat{\beta}_{MLE} = \frac{n}{x_n^\alpha} \quad (3.5)$$

따라서 본 논문에서는 형상(shape)모수  $\alpha$ 가 0.5, 1.5, 2(Rayleigh분포)인 경우를 선택하여 분석 비교한다.

### 3.2 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].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (3.6)$$

(4.6) 식에서  $T_R$ 과  $T_C$ 는 (2.1)식과 (2.6)식을 이용하면 다음 두 방정식에 의해서 계산 될 수 있다.

$$\ln R_0 = -\beta[(x + T_R)^\alpha - (T_R)^\alpha] \quad (3.7)$$

$$(c_1 - c_2)\alpha\beta T_C^{\alpha-1} + c_3 = 0 \quad (3.8)$$

단, 신뢰도  $R_0 = \hat{R}(x | T)$ .

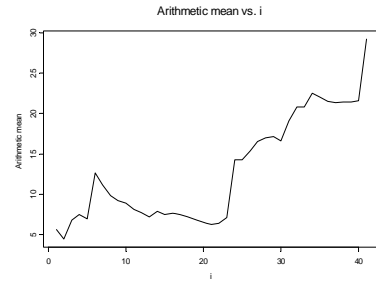
## 4. 수치적인 예

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 발생한 자료이며 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[3, 11].

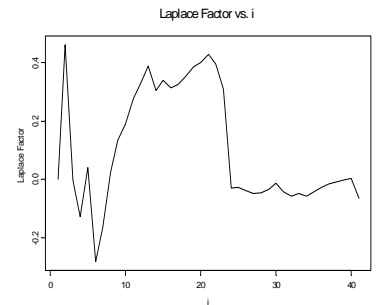
추세 분석에는 산술평균 검정(Arithmetic mean test)과 라플라스 추세 검정(Laplace trend test)등이 있다. 이 검정을 실시한 결과 [그림 2]에서 산술평균 검정결과 고장시간이 증가함에 따라 산술 평균이 거의 증가 추세를 보이고 있으므로 신뢰성장(Reliability growth) 속성을 가지고 있고 [그림 2]의 라플라스 추세 검정의 결과도 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도와 소프트웨어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

로그 포아송 실행시간모형의 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다.[17]. 그리고 와이블 특성 모형은  $\alpha=0.5$  일 때와  $\alpha=1.5, \alpha=2$ 일 때를 고정

하여 모수  $\beta$ 을 계산 한 값이 [표 1]에 요약되었다.



[그림 2] 산술평균검정



[그림 3] 라플라스 추세 검정

[표 1] 각 모형의 모수 추정값

Model	MLE	
Log Poission execution time	$\hat{\theta} = 0.020031$	$\hat{\lambda}_0 = 0.053067$
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 2] 최적 방출시간( $R_0 = 0.95$ )

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

[표 2]에서는  $c_1=5$ (\$),  $c_2=20$ (\$) 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 2]에 요약되었다.

이 표에서 최적방출시간은  $\alpha=0.5$ 인 경우는 근이 수렴되지 않아서 방출시간을 추정할 수 없었지만  $\alpha=1.5$ 인 경우는 기존의 로그 포아송 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.

## 5. 결 론

본 연구는 많은 분포들에 적합시키는 데 사용될 수 있는 와이블 분포를 적용한 무한고장 NHPP 모형을 이용하여 최적 방출시기에 관한 문제를 알아보았다. 즉, 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할수 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 와이블 분포를 적용한 방출시기 모형을 시도한 결과 와이블 분포 모형도 이 분야에서 적용 가능한 모형이 될수 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.(1981) "Statistical Models and Methods for Lifetime Data" John Wiley&Sons, New York.
- [2] 김희철(2005) "일반화감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한 연구", 한국컴퓨터정보학회논문지, 10권 6호, pp. 27-35.
- [3] S. S. Gokhale and K. S. Trivedi, (1999), "A time/structure based software reliability model", Annals of Software Engineering, 8, pp. 85-121.
- [4] 김희철(2006) "지수화 지수 분포에 의존한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구", 한국컴퓨터정보학회 논문지, 제11권 5호, pp. 9-18.
- [5] Musa, J. D and Okumoto, K.(1984.) "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Proc. the 7th Inter. Conf on Software Engineering. pp. 230-238,
- [6] 김대경(1988)"Musa-Okumoto의 대수 포아송 실행시간 모형에 근거한 비용-신뢰성 최적 정책", 품질경영학회지, 제26권 3호, pp. 141-149.
- [7] Almering, V. and Genuchten, M, V and Cloudt, G. and Sonnemans, P. J. M.(2007) " Using Software Reliability Growth Models in Practice ". IEEE SOFTWARE. pp. 82-88.
- [8] Xie, M. and Homg, G. Y.(1999) " Software release time determination based on unbound NHPP model". Proceeding of the 24th International Conference on Computers and Industrial Engineering. pp. 165-168.
- [9] Yang, B. and Xie. M.(2000) "A study of operational and testing reliability in software reliability analysis". RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol, 70, pp,323-329.
- [10] Huang, C. Y.(2005) "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.
- [11] Kuo, L. and Yang, T. Y. (1996) "Bayesian Computation of Software Reliability", Journal of the American Statistical Association, Vol.91, pp.763-773.
- [12] 간광연, 이재곤, 김희철(2005) "기록값 통계량 모형에 기초한 NHPP 소프트웨어 신뢰성에 관한 연구", 한국통신학회논문지,30권12T호,pp.356-363.
- [13] 김희철, 신현철, 김경수(2007)"기록값 통계량에 기초한 무한고장 NHPP 소프트웨어혼합 신뢰성장모형에 관한 연구", 정보, 보안 논문지, 제7권3호, pp. 51-60.
- [14] Pham, H. and Nordmann ,L. and Zhang, X (1999) "A General mperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate". IEEE Trans. on reliability, Vol, 48, No 2, pp, 169-175.
- [15] Musa, J. D, Iannino, A. and Okumoto, K.(1987) "Software Reliability: Measurement, Prediction, Application". McGraw Hill, New York.
- [16] K. Kanoun , J. C. Laprie (1996). Handbook of Software Reliability Engineering, M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, pp. 401-437.
- [17] 김희철(2008) "Log-Logistic 분포모형에 근거한 소프트웨어 최적방출시기에 관한 비교연구", 한국 컴퓨터정보학회 논문지, 제13권7호, pp. 1-9.