

결함 제거의 실패를 고려하는 소프트웨어 신뢰도 모델

金永輝* · 金成寅* · 李杭炯**

A Software Reliability Growth Model with Probability of Imperfect Debugging

Y.H. Kim*, S.I. Kim* and W.H. Lee**

Abstract

Common assumption we frequently encounter in early models of software reliability is that no new faults are introduced during the fault removal process. In real life, however, there are situations in which new faults are introduced as a result of imperfect debugging. This study alleviating this assumption by introducing the probability of perfect error-correction is an extension of Littlewood's work.

In this model, the system reliability, failure rates, mean time to failure and average failure frequency are obtained. Here, when the probability of perfect error-correction is one, the results appear identical with those of the previous studies. In the respect that the results of previous studies are special cases of this model, the model developed can be considered as a generalized one.

1. 서론

Littlewood[1]는 개발된 소프트웨어를 일정기간 (τ) 시험 운영하고 그 동안에 발생된 고장수(i)와 발생시각(t_1, t_2, \dots, t_i)에 의하여 모델의 모수들(parameters)를 추정하고, 이로부터 각종의 신뢰성 척도(measures)를 산출하는 모델을 제시하고 있다. 이 모델의 가정은 다음과 같다.

(i) 고장 발생율은 프로그램에 남아 있는 결점수에 비례한다.

(ii) 이 비례율은 시간이 지나면서 감소한다.

(iii) 고장을 발생시킨 결함은 완벽하게 제거된다.

Ohba와 Chou[2]는 위의 가정 중 마지막 가정의 비현실성을 지적하고 고장을 발생시킨 결함은 확률 $p=1-q$ ($0 \leq p \leq 1$)로 제거되는 상황을 가장 하였다. 이들은 제거되는 결함수(K)를 평균(ip)의 개념에서 보고 Littlewood의 모델($p=1$)을 분석한 결과, 동형구조의(isomorphic) 관계가 성립

* 고려대학교 공과대학 산업공학과

** 국방부 전력기획관실

함을 밝혔다.

이 논문에서는 확률 p 에 따라 제거되는 결함수 각각($K=1, 2, \dots, i$)의 확률을 사용하여 Littlewood의 모델을 확장한다. 이 모델의 각종 신뢰성 척도들에서 $p=1$ 인 특수한 경우가 Littlewood 모델의 결과이므로 이 모델은 Littlewood 모델을 일반화시킨 것으로 볼 수 있다.

2. 모델 및 신뢰성 척도

가정. 이 모델에서는 고장을 발생시킨 결함이 완벽하게 제거된다는 가정을 완화시킨다. 즉, 발견된 결함은 확률 $p(0 \leq p \leq 1)$ 로 제거된다. 따라서 발견된 총 결함 i 개 중에서 제거되는 결함수 K 는 확률변수이며 이것이 k 일 확률

$$P\{K=k\}$$

은 이항분포

$$\text{binomial}(i, p) = \begin{cases} \binom{i}{k} p^k (1-p)^{i-k}, & 0 \leq p \leq 1 \\ & k=0, 1, \dots, n \\ 0, & \text{그밖의 경우} \end{cases} \quad (1)$$

를 따른다.

이 밖의 가정은 Littlewood 모델에서와 같다. 즉,

- (i) 최초 소프트웨어 내에 총 N 개의 결함이 존재한다.
- (ii) 이들 결함들의 발생률, $\phi_1, \phi_2, \dots, \phi_N$ 은 확률변수이며 서로 독립이고 모수가 α 와 β 인 동일한 감마분포

$$\text{gamma}(\alpha, \beta) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} \phi^{\alpha-1} e^{-\beta\phi}, & \phi \geq 0 \\ 0, & \text{그밖의 경우} \end{cases} \quad (2)$$

를 따른다.

- (iii) 소프트웨어는 주어진 τ 시간동안 시험 운영되며 이 기간동안 i 개의 고장이 발생한다.

- (iv) 소프트웨어의 결함 발생률 $\Lambda = \phi_1 + \phi_2 + \dots + \phi_i (N-i \leq j \leq N)$ 가 값 λ 로 나타날 때 결함 발생시간은 지수분포

$$\exp(\lambda) = \begin{cases} \lambda e^{-\lambda t}, & t \geq 0 \\ 0, & \text{그밖의 경우} \end{cases} \quad (3)$$

를 따른다.

이들 가정으로부터 다음을 알 수 있다. 소프트웨어 내에 $j(N-i \leq j \leq N)$ 개의 결함이 존재할 때, 소프트웨어의 결함 발생률 $\Lambda = \phi_1 + \phi_2 + \dots + \phi_j$ 는 감마분포의 재생성질을 갖는 가법정리로부터 감마분포 $\text{gamma}(\alpha, \beta)$ 를 따른다.

고장발생률. 이제 시험 운영이 끝난 현재의 시점(시각 τ)에서 소프트웨어의 고장 발생률 Λ 의 분포를 구하여 본다. 이를 위하여는 먼저 시각 τ 에서 존재하고 있는 결함의 고장발생률의 확률밀도함수

$$g(\phi) = \{\phi \mid \text{기간}(0, \tau) \text{ 동안 결함 } j \text{에 의한 고장이 발생하지 않을 경우}\}$$

를 구하여야 한다. 이 결함이 기간 $(0, \tau)$ 동안에 고장을 발생시키지 않았다면 Littlewood의 결과처럼

$$g(\phi) = \text{gamma}(\alpha, \beta + \tau) \quad (4)$$

가 된다. 결함 j 가 시각 t 에서 고장을 발생시키고 제거되지 않은 경우, 문제를 간단하게 하기 위하여 시각 t 에서의 고장발생률이 감마분포

$$\text{gamma}(\alpha, \beta + t) \quad (5)$$

를 따른다고 가정한다. 이 가정하에서는 고장을 발생시켰으나 제거되지 않아서 시각 τ 에 남아있는 결함도 고장을 발생시키지 않아서 남아 있는 결함과 같이 그 고장발생률은 식 (4)의 $g(\phi) = \text{gamma}(\alpha, \beta + \tau)$ 를 따른다.

이 가정은 다음과 같은 면에서 타당성을 갖는다. 시간 t 에서 고장을 발생시킨 결함이 제거 노력에도 불구하고 제거되지 않았다면 그 고장발생률은 처음 보다는 줄어든 것이다. 식 (5)는 평균이 α/β 에서 $\alpha/(\beta+t)$ 로 감소된 감마분포를 따른다는 가정이다.

현재 시점에 $N-k$ 개의 결함이 존재할 확률은 $\text{binomial}(i, k)$ 이므로 현재 시점의 고장발생률

Λ 는 a 가 이항분포를 하는 감마분포 $gamma(a, \beta + \tau)$ 의 혼합분포(mixture distribution)를 따른다. 따라서 Λ 의 확률밀도 함수 $g(\lambda)$ 는

$$g(\lambda) = \sum_{k=1}^i binomial(i, k) \cdot gamma((n-k)a, \beta + \tau) \quad (6)$$

이다. 이 함수의 정식 표현(closed form)은 얻지 못하였으나 뒤에서 구한 것처럼 이를 이용한 고장 시간, 신뢰도, 고장율, 평균고장시간 등의 정식 표현은 가능하다.

그러나 τ 가 작은 초기 단계에서는 $N \gg i$ 이면 다음과 같은 정식표현을 얻는다. 즉, 등식

$$\begin{aligned} & \frac{1}{\Gamma((N-k)\alpha)} \\ &= \frac{1}{\Gamma((N-i)\alpha)} \\ & \frac{1}{[(N-k)\alpha-1] [(N-k)\alpha-2] \cdots [(N-i)\alpha]} \\ & \approx \frac{1}{\Gamma((N-i)\alpha)} \frac{1}{[(N-i)\alpha]^{(i-k)\alpha}} \\ &= \frac{1}{\Gamma((N-i)\alpha)} [(N-i)\alpha]^{-a(i-k)} \end{aligned}$$

을 이용하면

$$\begin{aligned} g(\lambda) &= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} \\ & \frac{(\beta + \tau)^{(N-k)\alpha}}{\Gamma((N-k)\alpha)} \lambda^{(n-k)\alpha-1} e^{-(\beta + \tau)\lambda} \\ &= \frac{1}{\Gamma((N-i)\alpha)} \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} \\ & \left[[(N-i)\alpha]^{-a} \right]^{(i-k)} (\beta + \tau)^{Na} \\ & \left[(\beta + \tau)^a \right]^k \lambda^{Na} (\lambda^{-a})^k \lambda^{-1} e^{-(\beta + \tau)\lambda} \\ &= \frac{(\beta + \tau)^{Na} \lambda^{Na-1} e^{-(\beta + \tau)\lambda}}{\Gamma((N-i)\alpha)} \\ & [p(\beta + \tau)^{-a} \lambda^{-a} + (1-p)[(N-i)\alpha]^{-a}]^i \end{aligned} \quad (7)$$

으로 정리된다.

물론 식 (6)에 $p=1$ 을 대입하면 Littlewood의 시스템 고장발생율에 대한 확률밀도함수

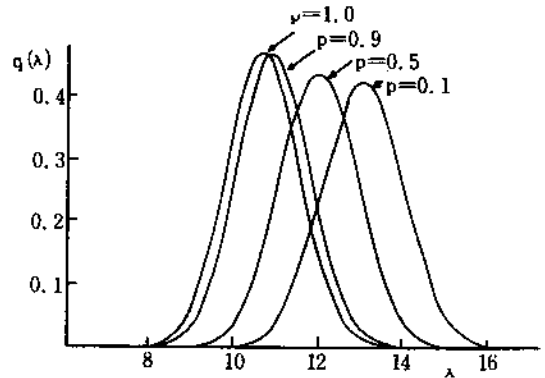


그림 1. 결함제거 확률 p 의 변화에 따른 고장발생율의 확률밀도 함수

($N=200, i=150, \tau=10, \alpha=1, \beta=5$).

$$g(\lambda) = \frac{(\beta + \tau)^{(N-i)\alpha}}{\Gamma((N-i)\alpha)} \lambda^{i(N-i)\alpha-1} e^{-(\beta + \tau)\lambda}$$

를 얻으며 근사식 (7)에 $p=1$ 을 대입하면 같은 결과를 얻음을 알 수 있다. 모수 α 와 β 의 특정 값에 대하여 p 를 변화시켜 보면서 고장발생율 확률밀도 함수를 살펴보면 그림 1과 같다.

그림 1은 결함제거 확률 p 가 적어짐에 따라 고장발생율이 증가함을 보여준다. 이는 결함을 완전하게 제거하지 못할 확률이 커짐에 따라 시스템 내에 잔존하는 결함의 수가 크게 됨을 나타내는 것이다. 따라서 본 모델이 불완전한 결함 제거 상황에서의 고장발생율의 변화를 적절히 반영함을 알 수 있다.

고장시간. 고장시간(time to failure)의 확률밀도 함수 $f(t)$ 는

$$\begin{aligned} f(t) &= \int_0^{\infty} \exp(-\lambda t) g(\lambda) d\lambda \\ &= \int_0^{\infty} \lambda e^{-\lambda t} \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} \\ & \frac{(\beta + \tau)^{(N-k)\alpha}}{\Gamma((N-k)\alpha)} \lambda^{(N-k)\alpha-1} e^{-(\beta + \tau)\lambda} d\lambda \end{aligned}$$

이다. 여기에서 \int 와 \sum 의 순서를 바꾸고 등식

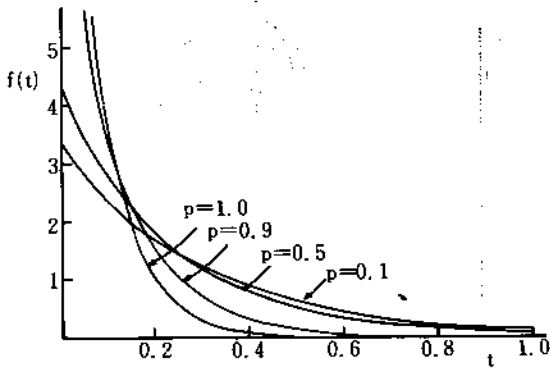


그림 2. 결합제거 확률 p 의 변화에 따른 고장시간의 확률밀도함수

($N=200, i=150, r=10, \alpha=1, \beta=5$).

$$\int_0^{\infty} \lambda^{(N-k)\alpha} e^{-(\beta+r+t)\lambda} d\lambda = \frac{\Gamma((N-k)\alpha+1)}{(\beta+r+t)^{(N-k)\alpha+1}}$$

을 이용하여 정리하면

$$f(t) = \frac{\alpha}{(\beta+r+t)} \left(\frac{\beta+r}{\beta+r+t} \right)^{N\alpha} \left[\sum_{k=0}^i \binom{i}{k} p^k q^{i-k} N \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha k} - \sum_{k=0}^i p^k q^{i-k} k \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha k} \right]$$

이 된다. 이 식에서 이항정리로부터 얻어지는 등식

$$\sum_{k=0}^i \binom{i}{k} p^k q^{i-k} N \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha k} = N \left[\left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} p + q \right]^i$$

와 잘 알려진 미분(d) 방법으로 얻어지는 등식

$$\begin{aligned} & \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} N \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha k-1} \\ &= \frac{d}{d \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha}} \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha k} \\ &= \frac{d}{d \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha}} \left[p \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} + q \right]^i \end{aligned}$$

$$= i p \left[p \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} \right]^{i-1}$$

을 대입하면

$$f(t) = \frac{\alpha}{(\beta+r+t)} \left(\frac{\beta+r}{\beta+r+t} \right)^{N\alpha} \left[N \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} p + q \right]^i - i p \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} \left(p \left(\frac{\beta+r+t}{\beta+r} \right)^{\alpha} + q \right)^{i-1}$$

을 얻는다.

이 식에 $p=1$ 을 대입하면

$$f(t) = \frac{(N-i)\alpha (\beta+r)^{(N-i)\alpha}}{(\beta+r+t)^{(N-i)\alpha+1}}$$

이어서 Littlewood의 결과를 얻는다.

그림 2는 p 를 변화시켜 보면서 고장시간과 확률밀도함수를 살펴본 것이다. p 가 큰 경우에는 결합이 많이 발생하게 되는 시험운행 기간 초기에 결합이 많이 제거되므로 확률밀도함수의 구배가 초기에 급격히 감소하는 경향을 보이고 있다. 반면 p 가 작아질수록 그 구배가 완만히 진행되는 경향을 보이고 있는데 이는 고장이 발생되었다고 하더라도 시스템의 잔존 결합수에 큰 영향을 미치지 못함을 반영하는 것이다.

신뢰도. 이제까지 시스템의 고장시간에 대한 확률밀도 함수 $f(t)$ 를 구하였으므로 시스템의 신뢰도, 고장율, 다음 고장까지의 기대시간(expected time to failure) 등을 구할 수 있다.

먼저, 시스템의 신뢰도는

$$\begin{aligned} R(t) &= 1 - F(t) \\ &= \int_t^{\infty} f(u) du \\ &= \int_t^{\infty} \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (N-k) \alpha \frac{(\beta+r)^{(N-k)\alpha}}{(\beta+r+u)^{(N-k)\alpha+1}} du \\ &= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (N-k) \alpha (\beta+r)^{(N-k)\alpha} \int_t^{\infty} \frac{1}{(\beta+r+u)^{(N-k)\alpha+1}} du \end{aligned}$$

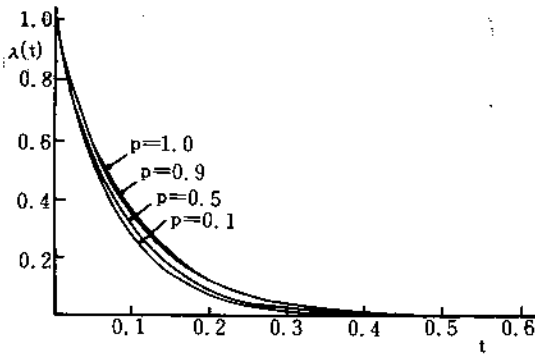


그림 3. 결함제거 확률 p 의 변화에 따른 신뢰도 함수 ($N=200, i=150, \tau=10, \alpha=1, \beta=5$).

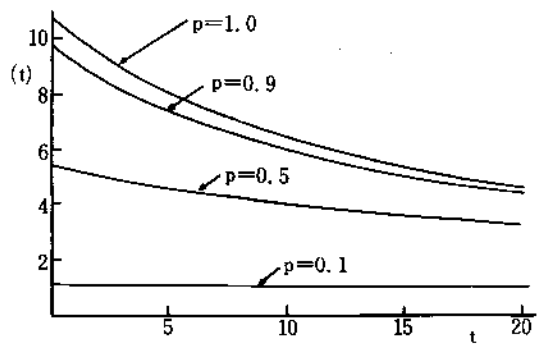


그림 4. 결함제거 확률 p 의 변화에 따른 고장율 ($N=200, i=150, \tau=10, \alpha=1, \beta=5$).

$$\begin{aligned} &= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (N-k)\alpha (\beta+\tau)^{(N-k)\alpha} \\ &\quad \frac{1}{(N-k)\alpha (\beta+\tau+t)^{(N-k)\alpha}} \\ &= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{(N-k)\alpha} \\ &= \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{Na} \left(\left(\frac{\beta+\tau+t}{\beta+\tau}\right)^a p + q\right)^i \end{aligned}$$

이 된다.

물론 이식에 $p=1$ 을 대입하면

$$R(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{(N-k)\alpha}$$

이 되어 Littlewood의 결과와 같다.

그림 3은 결함제거 확률 p 의 변화에 따른 신뢰도 함수 $R(t)$ 의 변화를 보여준다. 물론 p 가 작을 수록 신뢰도가 떨어짐을 설명하고 있다.

고장율. 시스템의 고장율(hazard rate) $\lambda(t)$ 는 다음과 같이 정리된다.

$$\begin{aligned} &\lambda(t) \\ &= \frac{f(t)}{R(t)} \\ &= \frac{\alpha}{(\beta+\tau+t)} \\ &\quad \left[N-i p \left(\frac{\beta+\tau+t}{\beta+\tau}\right)^a \left(\left(\frac{\beta+\tau+t}{\beta+\tau}\right)^a p + q\right)^{-1} \right] \end{aligned}$$

물론 $p=1$ 을 대입하면

$$\lambda(t) = \frac{(N-i)\alpha}{\beta+\tau+t}$$

이어서 Littlewood의 결과와 같다.

그림 4는 결함제거 확률 p 의 변화에 따른 고장율 함수를 보여준다. 앞에서와 마찬가지로 p 가 클 수록 시험운영이 진행됨에 따라 고장율이 현저히 낮아지며 반대의 경우에는 고장율이 그다지 낮아지지 않음을 알 수 있다. 특히 p 가 아주 작을 때에는 거의 일정한 고장율을 유지하게 되는데 이는 시험 운영이 진행되어도 소프트웨어의 품질향상에 기여하지 못하는 상황을 반영한다.

평균고장시간. 평균고장시간(MTTF: mean time to failure)은

$$\begin{aligned} &E[T] \\ &= \int_0^{\infty} [1-F(t)] dt \\ &= \int_0^{\infty} R(t) dt \\ &= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (\beta+\tau)^{(N-k)\alpha} \\ &\quad \int_0^{\infty} \left(\frac{1}{\beta+\tau+t}\right)^{(N-k)\alpha} dt \end{aligned}$$

$$= \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (\beta + \tau)^{(N-k)\alpha}$$

$$\frac{1}{(N-k)\alpha - 1} (\beta + \tau)^{-(N-k)\alpha + 1}$$

(여기서, $(N-k)\alpha > 1$ 이므로)

$$= (\beta + \tau) \sum_{k=0}^i \binom{i}{k} p^k q^{i-k} (\beta + \tau)^{(N-k)\alpha}$$

$$\frac{1}{(N-k)\alpha - 1}$$

가 된다. 이 식에 $p=1$ 을 대입하면

$$E[T] = (\beta + \tau) / [(N-i)\alpha - 1]$$

이어서 Littlewood의 결과를 얻는다.

그림 5는 결함제거 확률 p 의 변화에 따른 평균 고장시간을 보여준다.

3. 모수의 추정

본 모델은 Littlewood 모델의 3개의 모수 N, α, β 에 결함을 제대로 제거할 확률 p 를 추가하여 4개의 모수를 갖는다. 이들 모수들은 시험운영 기간 동안 나타난 고장시간의 간격

$$t_1, t_2, \dots, t_i$$

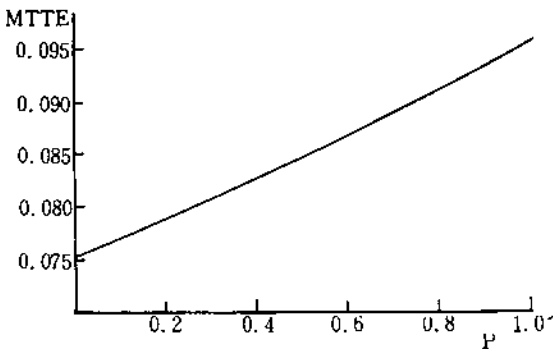


그림 5. 결함제거 확률 p 의 변화에 따른 평균고장 시간

($N=200, i=150, \tau=10, \alpha=1, \beta=5$).

로부터 추정하여야 한다. 이제 우도함수(likelihood function)를 구하여 보자. 먼저 m 번째 고장시간의 간격 t_m 의 확률밀도함수는

$$pdf\{t_m | t_1, t_2, \dots, t_{m-1}\}$$

$$= pdf\{t_m | \text{총 실행시간 } \sum_{j=1}^{m-1} t_j \text{에서}$$

($m-1$)개 고장 발생) $N\alpha$

$$= \left(\frac{\alpha}{\beta + \sum_{k=1}^{m-1} t_k + t_m} \right) \left(\frac{\beta + \sum_{j=1}^{m-1} t_j}{\beta + \sum_{j=1}^{m-1} t_j + t_m} \right)^{N\alpha}$$

$$\left[N \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha p + (1-p) \right]^{m-1}$$

$$- \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha$$

$$\left[p \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha + (1-p) \right]^{m-2} \quad (m-1)p$$

이다. 이로부터 우도함수는

$$L(\alpha, \beta, p, N)$$

$$= \prod_{m=1}^i \left(\frac{\alpha}{\beta + \sum_{j=1}^{m-1} t_j + t_m} \right) \left(\frac{\beta + \sum_{j=1}^{m-1} t_j}{\beta + \sum_{j=1}^{m-1} t_j + t_m} \right)^{N\alpha}$$

$$\left[N \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha p + (1-p) \right]^{m-1}$$

$$- \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha$$

$$\left[p \left(\frac{\beta + \sum_{j=1}^{m-1} t_j + t_m}{\beta + \sum_{j=1}^{m-1} t_j} \right)^\alpha + (1-p) \right]^{m-2} \quad (m-1)p \quad (8)$$

표 1. 시험운영 기간 $\tau=0.238$ 동안에 $N=60$, $\alpha=1.0$, $\beta=5.0$, $p=0.5$ 로부터 랜덤으로 생성된 $i=40$ 의 데이터

i	t_i	i	t_i	i	t_i	i	t_i
1	0.08333333	11	0.10000000	21	0.12500000	31	0.16666666
2	0.08474576	12	0.10204081	22	0.12820512	32	0.17241379
3	0.08620689	13	0.10416666	23	0.13157894	33	0.17857142
4	0.08771929	14	0.10638297	24	0.13513513	34	0.18518518
5	0.08928571	15	0.10869565	25	0.13888888	35	0.19230769
6	0.09090909	16	0.11111111	26	0.14285714	36	0.20000000
7	0.09259259	17	0.11363636	27	0.14705882	37	0.20833333
8	0.09433962	18	0.11627906	28	0.15151515	38	0.21739130
9	0.09615384	19	0.11904761	29	0.15625000	39	0.22727272
10	0.09803921	20	0.12195121	30	0.16129032	40	0.23809523

으로 구해진다.

물론 이 식에 $p=1$ 을 대입하면

$$L(\alpha, \beta, p, N)$$

$$= \prod_{m=1}^i \frac{(N-m+1)\alpha \left(\beta + \sum_{j=1}^{m-1} t_j\right)^{(N-m+1)\alpha}}{\left(\beta + \sum_{j=1}^m t_j\right)^{(N-m+1)\alpha+1}}$$

이어서 Littlewood의 결과를 얻는다.

식 (8)은 상당히 복잡하여 최우 추정량의 필요 조건인 각 수에 대하여 편미분한 식을 0으로 놓은 연립 방정식의 해를 구하기가 어렵다. 또한 Littlewood가 얻은 α 를 β, N 으로 표시한 식도 불가능하고 어떠한 형태로도 간략화되기가 어려운 것으로 보인다. 따라서 수치해석적인 방법을 통하여 이 식의 모양을 살펴보는 등의 경험적인 방법을 사용한다. 여기에는 Wolfram 등이 개발한 수학적 소프트웨어 패키지인 Mathematica가 유용하게 사용된다[3].

예제. 시험운영 기간 $\tau=0.238$ 동안에 <표 1>과 같은 $i=40$ 의 고장시간 간격이 관측되었을 때 모델의 모수인 N, α, β, p 를 추정하여 본다. 이 데이터는 실제로 $N=60, \alpha=1.0, \beta=5.0, p=0.5$ 로부터 랜덤으로 생성된 데이터이다.

먼저 $\alpha=1.5, p=0.3$ 으로 고정시킨 N, β 의 등고선은 그림 6과 같다. 등고선이 매끄러운 타원형

이 아니고 불규칙한 것은 계산점들의 간격이 넓기 때문이다. 그러나 이 간격을 좁히면 또 다른 곳에서 불규칙한 모양이 생긴다. 이 그림에서 최고점에서의 N, β 의 관계는 N 이 커질수록 β 가 커지는 선형의 관계를 볼 수 있다. α 와 p 를 다른 값으로 고정시킨 경우에도 모두 이러한 선형 관계를 볼 수 있다. 이는 고장발생율의 평균

$$\alpha(N-j)/\left(\beta + \sum_{j=1}^i t_j\right), \quad j=1, 2, \dots, i$$

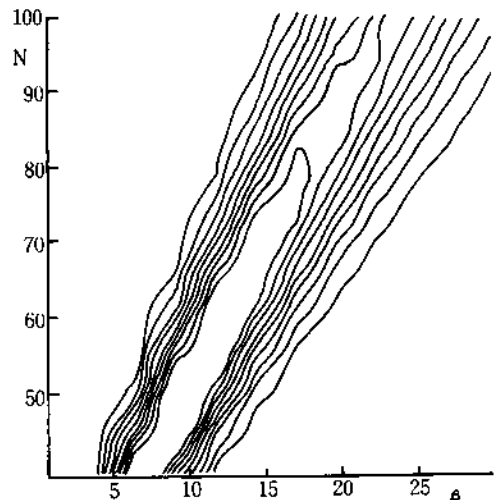


그림 6. N, β 의 우도함수의 등고선 ($\alpha=1.5, p=0.3$).

이 일정한 값을 유지하려는 것으로부터 짐작할 수 있다. 몇개의 등고선으로부터 이 관계를

$$\beta = -14.5 + 0.3N \quad (9)$$

으로 놓았다.

최종적으로 N, β 의 선형관계식인 식 (9)와 N, α 의 선형관계식인 식(10)을 유지시킨 N, p 의 우

도함수의 3차원 그래프와 그 등고선은 각각 [그림 7], 그림 8과 같다.

이들 그림으로부터 우도함수를 최대로 하는 점은 $N=64, \alpha=0.959, \beta=4.719, p=0.4$ 로 추정된다. 이는 본래 이 관측치가 생성된 모집단의 모수와 비슷하게 일치하는 결과이다.

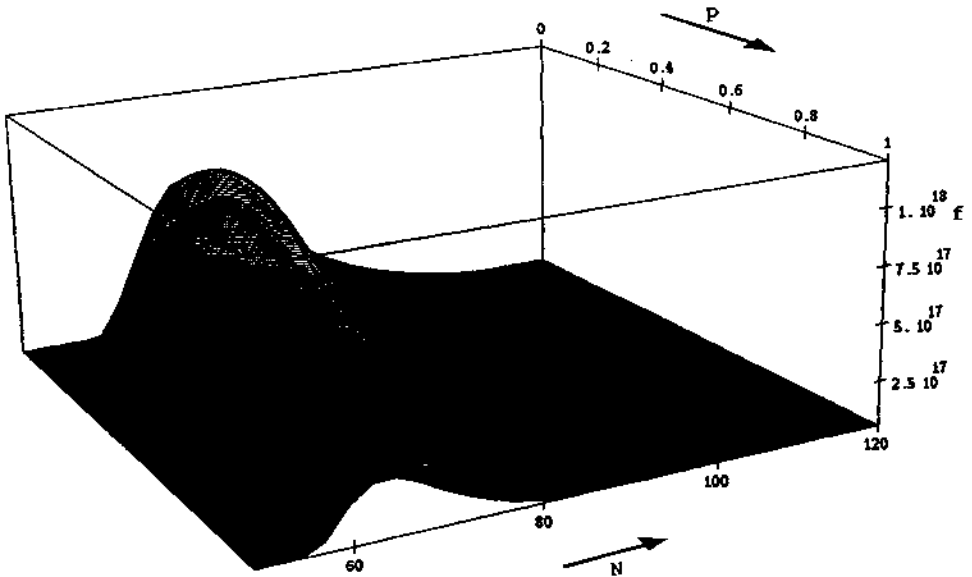


그림 7. N, p 의 우도함수
 $(\beta = -14.5 + 0.3N, \alpha = 3.778 - 0.44N).$

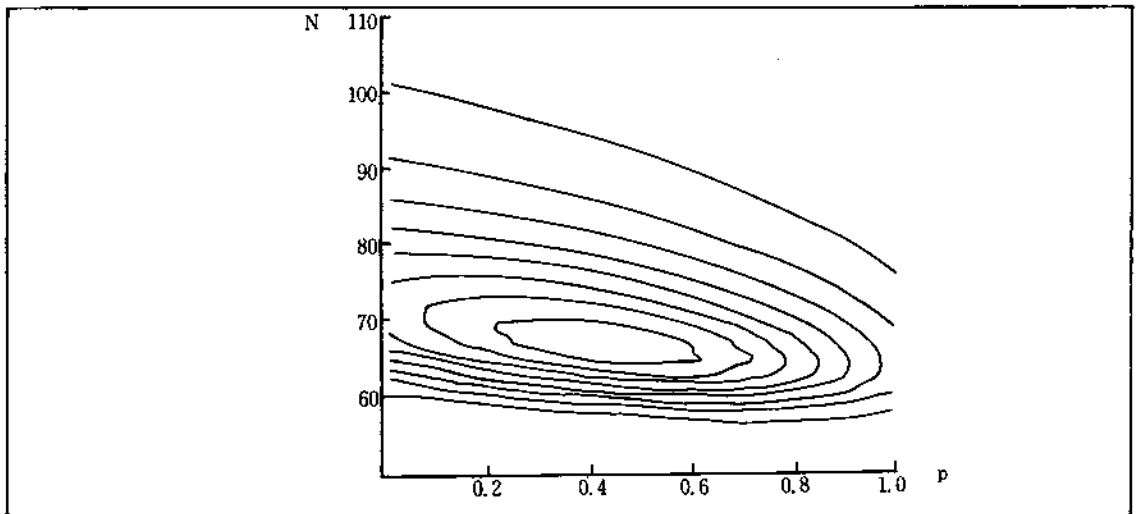


그림 8. N, p 의 우도함수의 등고선
 $(\beta = -14.5 + 0.3N, \alpha = 3.778 - 0.44N).$

4. 결 론

참 고 문 헌

기존의 소프트웨어 신뢰도 모델은 고장발생시 그 원인이 되는 결함을 항상 완벽하게 제거할 수 있다고 가정한다. 그러나 소프트웨어에서는 결함의 수정이 완벽하지 못한 상황이 흔히 발생한다. 본 연구에서는 고장발생시 그 결함이 올바르게 수정되어 완전히 제거되는 경우와 결함이 수정되지 못하거나 또 다른 결함을 범하게 되는 두 상황을 반영할 수 있는 모델을 수립하였다. 두 상황 중 어느 하나의 상황이 일어난 가능성을 확률 p 로 나타냄으로써 기존의 모델은 $p=1$ 의 특수한 경우가 된다.

이 모델에서 고장 발생율, 시스템의 신뢰도, 고장시간, 고장율, 평균 고장시간 등을 구하고, 결함이 완벽하게 수정되는 확률 p 가 이들에게 주는 영향을 분석하여 보았다. 이 확률을 1로 두면 기존 연구의 결과와 같게 되므로 기존의 연구는 본 연구의 특정 경우로 나타난다. 따라서 본 연구는 기존 연구의 일반화시킨 것이다.

Littlewood에서와 같이 최우추정법에 의한 모수 추정을 시도하였으나 만족스럽지 못하고 다른 방법을 시도하여 볼 필요가 있다. 또한 본 연구의 결과를 이용하여 소프트웨어 시스템의 최적 상품화 시기를 결정하는 연구를 생각할 수 있으며, 고장을 수정하는데 소요되는 시간을 고려하고, 잘못 수정하여 여러개의 결함이 추가되는 보다 일반화된 모델의 발전 등은 앞으로의 과제가 된다.

[1] B. Littlewood, "Stochastic Reliability Growth: A Model for Fault Removal in Computer Programs and Hardware Designs," *IEEE Transactions on Reliability*, Vol. R-30, pp. 313-320, 1981.

[2] M. Ohba and X. Chou, "Does Imperfect Debugging Affect Software Reliability Growth?" *Proceedings of 11th International IEEE Conference on Software Engineering*, pp. 237-244, 1989.

[3] S. Wolfram, *Mathematica*, Addison Wesley Publishing, Company, Inc., Redwood City, CA., 1988.