International Journal of Reliability and Applications
Vol. 2, No. 3, pp. 147-160, 2001

Invited Paper

Cost Implications of Imperfect Repair in Software Reliability

Philip J. Boland
Department of Statistics
National University of Ireland - Dublin, Belfield, Dublin 4, Ireland

Nóra Ní Chuív *
Department of Mathematics and Statistics
University of New Brunswick, Fredericton, New Brunswick, E3B 5A3, Canada

Abstract. The reliability of computer software is of prime importance for all developers of software. The complicated nature of detecting and removing faults from software has led to a plethora of models for reliability growth. One of the most basic of these is the Jelinski Moranda model, where it is assumed that there are N faults in the software, and that in testing, bugs (or faults) are encountered (and removed when detected) according to a stochastic process at a rate which at a given point in time is proportional to the number of bugs remaining in the system. In this research, we consider the possibility that imperfect repair may occur in any attempt to remove a detected bug in the Jelinski Moranda model. We let p represent the probability that a fault which is discovered or detected is actually perfectly repaired. The possibility that the probability p may differ before and after release of the software is also considered. The distribution of both the number of bugs detected and perfectly repaired in a given time period is studied. Cost models for the development and release of software are investigated, and the impact of the parameter p on the optimal release time minimizing expected costs is assessed.

Key Words : *Software Testing, Jelinski Moranda Model, Software Reliability, Optimal Release Time, Birth Processes, Cost Models, Imperfect Repair.*

1. INTRODUCTION

The often complicated nature of detecting and removing faults in computer software has led to an extensive literature on software reliability modeling. The books by Lyu (in particular the chapter by Farr(1996)), Singpurwalla and Wilson (1999),

* *E-mail address : nora@math.unb.ca*

and Musa, Ianinno and Okumoto (1987) provide excellent overall references for much of the recent developments in software reliability modeling, and the survey by Xie (2000) provides a good recent overview of the subject. One of the earliest and most basic models in software reliability is however the Jelinski Moranda model which was introduced in 1972. In this model it assumed that there are N faults in the software, and that in testing, bugs (or faults) are encountered (and removed when detected) according to a stochastic process at a rate which at a given point in time is proportional to the number of bugs remaining in the system. Hence if at a given point in time r faults have been encountered and removed, then the failure rate for the software is of the form $\Lambda(N-r)$ for some constant Λ. This is of course equivalent to assuming that the time T_{r+1} between the r^{th} and $(r+1)^{st}$ failures is exponentially distributed with parameter (failure rate) $\Lambda(N-r)$ for $r = 0, 1, \ldots, N-1$. Because this model makes the implicit assumptions that all of the bugs remaining in the software at a given time contribute equally to the failure rate, and that when bugs are encountered they are perfectly repaired or removed, there have been many generalizations and extensions of this model. Many of these generalizations assume that repair of a fault when it is encountered is perfect, although there are some exceptions. Mazzuchi and Soyer (1988) introduced a model whereby the failure rate Λ_{r+1} between the r^{th} and $(r+1)^{st}$ failures is a random variable which is stochastically decreasing. This allows the possibility that repair of faults is not perfect and in particular that the failure rate in practice might actually occasionally increase. Goel and Okumoto (1978) introduced a generalization whereby the failure rate between the r^{th} and $(r+1)^{st}$ failure is of the form $\Lambda(N-pr)$. The model we introduce here allows for the imperfect repair of faults whereby when a fault is detected and not correctly fixed, then the failure rate for the occurrence of the next fault does not change. Let us assume that p is the probability that a failure (bug) which is detected is perfectly (correctly) repaired, and hence the failure rate for the software between the *detection and perfect repair* of the r^{th} and $(r+1)^{st}$ failures is $\Lambda p(N-r)$. However, in between such events there might be discoveries of faults which are subsequently not perfectly repaired (and hence do not change the failure rate of the software as a whole). In this paper we study cost structures for this model and discuss optimal stopping times for release of software. Of particular interest is how the probability of perfect repair p influences the optimal release time for software.

The concept of perfect and imperfect repair for engineering systems has been extensively treated in the reliability literature. Brown and Proschan (1983) were among the first to deal with the possibility that the repair of a system might be perfect with probability p (and otherwise that an imperfect or *minimal repair* is made in the sense that the system repair rate returns to what it was just prior to failure), and they studied the distribution between renewal states for systems. Some other useful references for imperfect repair of systems are Finkelstein (1997), Lim and Park (1999), Makis and Jardine (1992), and Sheu (1998).

2. THE NUMBERS OF SOFTWARE FAULTS REMOVED $R(t)$ AND DETECTED $M(t)$ IN THE INTERVAL $(0, t]$

As in the Jelinski-Moranda model, we shall assume that initially there are N faults in the software each with failure rate Λ, resulting in an initial failure rate for the system of ΛN. When a fault is detected in the software, we assume it is perfectly repaired with probability p, in which case the number of faults in the software is reduced by 1 (and consequently the failure rate is reduced by Λ). Otherwise the number of bugs in the software (and the failure rate for the software) remains the same. For any interval of time $(0, t]$, we let $M(t)$ be the total number of faults detected or met by time t, and $R(t)$ be the number of bugs which have been removed (that is detected and perfectly repaired) by time t. The number of *unsuccessful* repairs in the interval $(0, t]$ will be denoted by $U(t) = M(t) - R(t)$. In the standard Jelinski Moranda model the times between removals of faults are independent (but not identically distributed) exponential random variables, while in our model (due to the possibility of the detection of a fault which is not removed) the times between removals of faults are independent compound Geometric random variables. However times between detections (which are not necessarily removals) of faults are still independent exponential random variables. For any integers u and r (where $0 \leq r \leq N$) we let

$$P_r(t) = \text{Prob}(R(t) = r) \quad \text{and} \quad P_{u,r}(t) = \text{Prob}(U(t) = u, R(t) = r), \tag{2.1}$$

We will derive useful expressions for these quantities in terms of t and the relevant parameters Λ, N and p. Using a standard differential equations approach, it may be established that for any $1 \leq r \leq N$

$$P_r'(t) = -\Lambda p(N-r)P_r(t) + \Lambda p(N-r+1)P_{r-1}(t).$$

With the initial conditions that $P_r(0) = 0$ for $1 \leq r \leq N$ and $P_0(t) = e^{-\Lambda pNt}$ for $t \geq 0$, one may establish (by induction or otherwise) that

$$P_r(t) = \binom{N}{r} e^{-\Lambda p(N-r)t}(1 - e^{-\Lambda pt})^r. \tag{2.2}$$

Note that $R(t)$ is a *binomial* birth process (that is $R(t)$ is $Bin(N, 1 - e^{-\Lambda pt})$ for any $t > 0$), and that when $p = 1$ (perfect repair) we obtain the standard Jelinski Moranda model. We will now derive an expression for $P_{u,r}(t)$ using an integral approach. The following technical results will be useful and can be proved by induction:

Lemma 1 *For any non-negative integers r, u and nonnegative α, x*

$$\Lambda^r \int_0^x \int_0^{x_r} \cdots \int_0^{x_2} \frac{[\alpha + \Lambda x_r + \Lambda x_{r-1} + \cdots + \Lambda x_1]^u}{u!} e^{-[\alpha + \Lambda x_r + \cdots \Lambda x_1]} dx_1\, dx_2 \ldots dx$$

$$= \frac{1}{r!} \sum_{j_1=0}^{u} \sum_{j_2=0}^{j_1} \cdots \sum_{j_r=0}^{j_{r-1}} \sum_{i=0}^{r} \binom{r}{i} (-1)^i \frac{[\alpha + i\Lambda x]^{j_r}}{j_r!} e^{-[\alpha + i\Lambda x]}. \tag{2.3}$$

Lemma 2 *For all $n, j \geq 0$*

$$\sum_{k=0}^{n} \frac{(k+j)!}{k!j!} = \frac{(n+j+1)!}{n!(j+1)!}. \tag{2.4}$$

Now let $x_1 \leq x_2 \leq \cdots \leq x_r$ be the occurrence times of the r bugs which are detected and successfully removed in the time interval $(0, t]$. Let $u_0, u_1, \ldots, u_r$ be the number of bugs which are encountered but *not* removed in the time intervals $(0, x_1], (x_1, x_2], \ldots, (x_r, t]$. For convenience we will furthermore set $x_0 = 0$ and $x_{r+1} = t$. Letting $\mathbf{x} = (x_0, x_1, x_2, ...x_r)$ and $\mathbf{u} = (u_0, u_1, ...u_r)$ we have that the joint density function for $\mathbf{x}$ and $\mathbf{u}$ is given by

$$\begin{aligned} f(\mathbf{x}, \mathbf{u}) &= p^r \prod_{i=0}^{r} \left\{ \frac{[\Lambda(N-i)(x_{i+1}-x_i)]^{u_i}}{u_i!} e^{-\Lambda(N-i)(x_{i+1}-x_i)} (1-p)^{u_i} \right\} \prod_{i=0}^{r-1} \Lambda(N-i) \\ &= p^r (1-p)^{\sum_{i=0}^{r} u_i} \Lambda^r \frac{N!}{(N-r)!} (e^{-\sum_{i=0}^{r} \Lambda(N-i)(x_{i+1}-x_i)}) \left\{ \prod_{i=0}^{r} \frac{[\Lambda(N-i)(x_{i+1}-x_i)]^{u_i}}{u_i!} \right\} \\ &= p^r (1-p)^{\sum_{i=0}^{r} u_i} \Lambda^r \binom{N}{r} r! (e^{-\Lambda[(N-r)t + x_r + \cdots + x_1]}) \prod_{i=0}^{r} \frac{[\Lambda(N-i)(x_{i+1}-x_i)]^{u_i}}{u_i!}. \end{aligned}$$

On integration it follows that

$$\begin{aligned} P_{u,r}(t) &= \sum_{u_0 + \cdots u_r = u} \int_0^t \int_0^{x_r} \cdots \int_0^{x_2} f(\mathbf{x}, \mathbf{u}) \mathbf{dx_1}\, \mathbf{dx_2} \ldots \mathbf{dx_r} \\ &= \Lambda^r \binom{N}{r} r! p^r (1-p)^u \int_0^t \int_0^{x_r} \cdots \int_0^{x_2} e^{-\Lambda[(N-r)t + x_r \cdots + x_1]} \\ &\quad \cdot \left\{ \frac{1}{u!} \sum_{u_0 + \cdot + u_r = u} \prod_{i=0}^{r} \frac{u! [\Lambda(N-i)(x_{i+1}-x_i)]^{u_i}}{u_i!} \right\} dx_1 \ldots dx_r. \end{aligned}$$

Recognizing the multinomial expansion

$$\sum_{u_0 + \cdots + u_r = u} \prod_{i=0}^{r} \frac{u! [\Lambda(N-i)(x_{i+1}-x_i)]^{u_i}}{u_i!} = \left[\sum_{i=0}^{r} \Lambda(N-i)(x_{i+1}-x_i) \right]^u$$

the expression for $P_{u,r}(t)$ simplifies to

$$\begin{aligned} P_{u,r}(t) &= \binom{N}{r} p^r (1-p)^u \Lambda^r r! \\ &\quad \cdot \int_0^t \int_0^{x_r} \cdots \int_0^{x_2} \frac{[\Lambda(N-r)t + \Lambda x_r + \cdots + \Lambda x_1]^u}{u!} e^{-\Lambda[(N-r)t + x_r + \cdots + x_1]} dx_1 \cdots dx_r \end{aligned}$$

Using Lemma 1, these iterative integrals are replaced by an iterative sum

$$P_{u,r}(t) = \binom{N}{r} p^r (1-p)^u \sum_{j_1=0}^{u} \sum_{j_2=0}^{j_1} \cdots \sum_{j_r=0}^{j_{r-1}} \sum_{i=0}^{r} \binom{r}{i} (-1)^i \frac{[\Lambda(N-r)t + i\Lambda t]^{j_r}}{j_r!} e^{-\Lambda[(N-r)t + it]}$$

$$= \binom{N}{r}p^r(1-p)^u \sum_{i=0}^{r} \sum_{u-j_r=0}^{u} \binom{r}{i}(-1)^i e^{-\Lambda[(N-r)t+it]} \frac{[\Lambda(N-r)t+i\Lambda t]^{j_r}}{j_r!} \sum_{u-j_{r-1}=0}^{u-j_r} \cdots \sum_{u-j_1=0}^{u-j_2} 1$$

$$= \binom{N}{r}p^r(1-p)^u \sum_{i=0}^{r} \binom{r}{i}(-1)^i e^{-\Lambda(N-r+i)t} \sum_{j=0}^{u} \frac{[\Lambda(N-r+i)t]^j}{j!} \frac{(r+u-j-1)!}{(r-1)!(u-j)!}. \qquad (2.5)$$

(repeated applications of Lemma 2 show that

$$\sum_{u-j_{r-1}=o}^{u-j_r} \cdots \sum_{u-j_1=0}^{u-j_2} 1 = \sum_{N_{r-1}=0}^{N_r} \cdots \sum_{N_1=0}^{N_2} 1 = \frac{(N_r+r-1)!}{(r-1)!N_r!} = \frac{(r+u-j_r-1)!}{(r-1)!(u-j_r)!}$$

where $N_l = u - j_l$). Complicated as it is, equation (2.5) gives us a precise expression for the probability $\text{Prob}(U(t) = u, R(t) = r)$. Note that summing equation (2.5) over u yields $P_r(t)$, confirming the result obtained previously. We have seen that $E[R(t)] = N(1-e^{-\Lambda pt})$, and we now derive a similar expression for $E[U(t)]$.

$$\begin{aligned}
E[U(t)] &= \sum_{r=0}^{N}\sum_{u=0}^{\infty} uP_{u,r}(t) \\
&= \sum_{r=0}^{N}\binom{N}{r}p^r \sum_{i=0}^{r}\binom{r}{i}(-1)^i e^{-\Lambda(N-r+i)t} \sum_{u=0}^{\infty}\sum_{j=0}^{u} \frac{(r+u-j-1)!}{(r-1)!(u-j)!}\frac{[\Lambda(N-r+i)t]^j}{j!} u(1-p)^u \\
&= \sum_{r=0}^{N}\binom{N}{r}p^r \sum_{i=0}^{r}\binom{r}{i}(-1)^{i+1} e^{-\Lambda(N-r+i)t} \\
&\quad \cdot(1-p)\frac{\partial}{\partial p}\left\{\sum_{j=0}^{\infty}\frac{[\Lambda(N-r+i)t]^j}{j!}(1-p)^j \sum_{u-j=0}^{\infty}\frac{(r+u-j-1)!}{(r-1)!(u-j)!}(1-p)^{u-j}\right\} \\
&= (1-p)\sum_{r=0}^{N}\binom{N}{r}p^r \sum_{i=0}^{r}\binom{r}{i}(-1)^{i+1} e^{-\Lambda(N-r+i)t}\frac{\partial}{\partial p}\left\{e^{\Lambda(1-p)(N-r+i)t}p^{-r}\right\} \\
&= (1-p)\sum_{r=0}^{N}\binom{N}{r}p^r \sum_{i=0}^{r}\binom{r}{i}(-1)^{i} e^{-\Lambda(N-r+i)t}\{\Lambda(N-r+i)t + rp^{-1}\}e^{\Lambda(1-p)(N-r+i)t}p^{-r} \\
&= (1-p)\sum_{r=0}^{N}\binom{N}{r}e^{-\Lambda p(N-r)t} \sum_{i=0}^{r}\binom{r}{i}(-1)^{i} e^{-\Lambda pit}\left\{\Lambda(N-r+i)t + \frac{r}{p}\right\} \\
&= (1-p)\sum_{r=0}^{N}\binom{N}{r}e^{-\Lambda p(N-r)t}\left\{\left[\Lambda(N-r)t+\frac{r}{p}\right]\sum_{i=0}^{r}\binom{r}{i}(-1)^i e^{-\Lambda pti} + \Lambda t\sum_{i=0}^{r} i\binom{r}{i}(-1)^i e^{-\Lambda pti}\right\} \\
&= (1-p)\sum_{r=0}^{N}\binom{N}{r}e^{-\Lambda p(N-r)t}\left\{\left[\Lambda(N-r)t+\frac{r}{p}\right](1-e^{-\Lambda pt})^r - \Lambda rte^{-\Lambda pt}(1-e^{-\Lambda pt})^{r-1}\right\} \\
&= (1-p)\{\Lambda Nte^{-\Lambda pt} + \frac{1}{p}N(1-e^{-\Lambda pt}) - \Lambda Nte^{-\Lambda pt}\} \\
&= \frac{1-p}{p}N(1-e^{-\Lambda pt}). \qquad (2.6)
\end{aligned}$$

Our expression for $E[U(t)]$ may also be established by induction on N and conditioning on the time X_1 until the first bug is detected and perfectly repaired.

Although there does not seem to be an easy verification of (2.6), the result has a certain rationale to it. The number of bugs encountered and not successfully repaired between each successful repair is *Geometric* with parameter p (and mean $(1-p)/p$) and $E[R(t)] = N(1 - e^{-\Lambda pt})$. $U(t)$ may be expressed as a sum of $R(t)$ geometric random variables with parameter p plus a remainder term (# unsuccessful repairs in the interval $(x_{R(t)}, t]$). Given that $M(t) = R(t) + U(t)$, it is clear that $E[M(t)] = (N/p)(1 - e^{-\Lambda pt})$.

3. OPTIMAL RELEASE TIME FOR TESTING SOFTWARE

Considerable development and testing is usually involved before a piece of software is made commercially available. What is the optimal release time T^* for a piece of software? This is of course a crucial question of considerable (financial and otherwise) consequence. Often software is introduced knowing that it possesses some (many?) errors of a minor nature, but hopefully without any which are fundamental to the functioning of the system. Competition often necessitates that software is released quicker than it might otherwise be. The problem of determining the optimal release time for software is an important one that has been extensively treated in the literature. Okumoto and Goel (1980) introduce a basic cost model and determine the optimal release time for their time dependent error detection model. An interesting generalization of this model was studied by Wang and Pham (1996) in which they extend the cost model to incorporate costs for warranty periods, error removals and risk costs. Singpurwalla (1991) addresses finding an optimal release time using the Jelinski Moranda model with a Bayesian decision theoretic approach. Boland and Singh (2000b) determine optimal release times for the Moranda geometric model for software failures (see also Moranda (1975)). Other approaches to determining the optimal release time which are of a decision theoretic nature are treated in McDaid (1998) and Dalal and Mallows (1988). We consider two cost structure models for the release of software at time T for our imperfect repair model, and determine the time T^* which minimizes expected costs. An important part of our considerations is to evaluate the impact of the parameter p (which represents the probability of perfect software repair) on the optimal release time T^*.

In our cost models we will consider two distinct cases. In the first case we consider a fixed software *life-cycle* time t_0 where it is desired that if the software is released at time T, it functions well in the period $(T, t_0]$. In the second case we will want the software to function for some fixed *mission* time τ after release (that is to say, in the period $(T, T + \tau]$). Let us use c_1 to be the cost associated with encountering a bug (whether successfully repaired or not) during the testing period $(0, T]$, c_2 to be the cost of dealing with a bug encountered after the release time T, and c_3 to be the cost of testing per unit time during the testing period. A common cost function model used for releasing the software at time T (see Okumoto and Goel (1980), McDaid (1998), and Boland and Singh (2000b)) takes the form

$$C(T) = c_1 M(T) + c_2[M(g(T)) - M(T)] + c_3 T \tag{3.1}$$

where $g(T) = g_1(T) = t_0$ in case 1 where we are considering a fixed life cycle time t_0, and $g(T) = g_2(T) = T + \tau$ in case 2 when we consider a fixed mission time τ. The total expected cost for releasing the software at time T therefore takes the form

$$\begin{aligned} E[C(T)] &= c_1 E[M(T)] + c_2(E[M(g(T))] - E[M(T)]) + c_3 T \\ &= c_2 \frac{N}{p}(1 - e^{-\Lambda p g(T)}) - (c_2 - c_1)\frac{N}{p}(1 - e^{-\Lambda p T}) + c_3 T. \end{aligned} \quad (3.2)$$

Differentiating with respect to T, we find unique critical points;

$$T = \frac{1}{\Lambda p} \ln\left[\frac{\Lambda N(c_2 - c_1)}{c_3}\right], \quad (3.3)$$

for case 1, while in case 2 the solution is

$$T = \frac{1}{\Lambda p} \ln\left[\frac{\Lambda N(c_2(1 - e^{-\Lambda p \tau}) - c_1)}{c_3}\right]. \quad (3.4)$$

These give a minimum if $\Lambda N[c_2 - c_1] > c_3$ in case 1, and $\Lambda N[c_2(1 - e^{-\Lambda p \tau}) - c_1] > c_3$ in case 2. Otherwise $\frac{\partial}{\partial T} E[C(T)] > 0$, and T should be taken as zero. Furthermore we must have $T \leq g(T)$, thus, denoting the optimal choice of T by T^*,

$$T^* = min\left(t_0, max\left(0, \frac{1}{\Lambda p} \ln\left[\frac{\Lambda N(c_2 - c_1)}{c_3}\right]\right)\right) \quad (3.5)$$

in the case of the fixed life cycle t_0, and for the case of a fixed mission time τ,

$$T^* = max\left(0, \frac{1}{\Lambda p} \ln\left[\frac{\Lambda N[c_2(1 - e^{-\Lambda p \tau}) - c_1]}{c_3}\right]\right). \quad (3.6)$$

Up to this point we have assumed that N, the number of bugs originally in the system, is fixed. We now consider the situation where N is itself a random variable. Some common distributions used to model N are: $N \sim Poisson(\theta)$, $N \sim NegBin(\kappa, \theta)$ and $N \sim Bin(M, \theta)$. In order to minimize the expected cost of releasing the software, we need expressions for $E[R(t)]$,the expected number of bugs encountered and removed in $(0, t]$ and $E[U(t)]$, the expected number of bugs encountered, but not removed in $(0, t]$. Conditioning on N and using our previous results, it follows that

$$E[R(t)] = E_N[E[R(t|N)]] = E[N](1 - e^{-\Lambda p t})$$

and

$$E[U(t)] = E_N[E[U(t|N)]] = E[N][\frac{1-p}{p}(1 - e^{-\Lambda p t})].$$

Consequently we obtain $E[M(t)] = E[N]\frac{1}{p}(1 - e^{-\Lambda p t})$ and thus as with the derivation of (3.2)

$$E[\text{C(T)}] = c_2 \frac{E[N]}{p}(1 - e^{-\Lambda p g(T)}) - (c_2 - c_1)\frac{E[N]}{p}(1 - e^{-\Lambda p T}) + c_3 T. \quad (3.7)$$

Proceeding as in the case where N is known, one may establish formulae for the optimal T identical to those derived in (3.5) and (3.6) with N replaced with $E[N]$.

4. NUMBER OF FAULTS REMAINING AT TIME T

Of considerable concern is the number of faults or bugs $N - R(T)$ remaining in the system at any time and in particular at release time T. In the situation where N is known, we have seen that $R(T) \sim Bin(N, 1 - e^{-\Lambda pt})$ and consequently that $N - R(T) \sim Bin(N, e^{-\Lambda pt})$. More generally suppose that N is random with distribution π. Therefore

$$\begin{aligned} P(N - R(T) = j) &= \sum_{n=j}^{\infty} P(N - R(T) = j \mid N = n)\pi(n) \\ &= \sum_{n=j}^{\infty} P_{n-j}(T)\pi(n) \\ &= \sum_{n=j}^{\infty} \binom{n}{j} e^{-\Lambda pTj}(1 - e^{-\Lambda pT})^{n-j}\pi(n). \end{aligned} \tag{4.1}$$

The distribution of $N - R(T)$ often takes the same form as that of N, as Table 1 demonstrates:

5. DIFFERENT PROBABILITIES OF REMOVING BUGS

In some situations it might be reasonable to assume that the probability of a perfect repair p changes after the software is released at time T. Let us therefore assume that p_1 is the probability that a bug encountered during the testing period $(0, T]$ is removed (perfectly repaired) and that p_2 is the corresponding probability for a bug encountered after release at time T. Using the same cost model for testing the software and by conditioning on the number of bugs $N - R(T)$ left in the system at time T, one may show that

$$E[C(T)] = c_1 \frac{N}{p_1}(1 - e^{-\Lambda p_1 T}) + c_2 \frac{N e^{-\Lambda p_1 T}}{p_2}(1 - e^{-\Lambda p_2 (g(T) - T)}) + c_3 T. \tag{5.1}$$

In the case of a fixed life-cycle time t_0 we find the optimal (minimizing expected costs) release time by solving the equation

$$(c_1 \Lambda N - c_2 \Lambda N \frac{p_1}{p_2})\ e^{-\Lambda p_1 T} + c_2 \Lambda N \frac{p_1 - p_2}{p_2}\ e^{-\Lambda p_2 t_0} e^{-\Lambda(p_1 - p_2)T} + c_3 = 0 \tag{5.2}$$

Since this equation cannot be solved in closed form, the value of T^* must be found numerically. Moreover in this case it is not immediately obvious whether there is a unique solution to (5.2).

In the case of a fixed mission time τ, one is able to solve explicitly the equation $E'(C(T)) = 0$, therefore obtaining the optimal release time to be

$$T^* = \max\left(0, \frac{1}{\Lambda p_1} \ln\left[\frac{\Lambda N(c_2 \frac{p_1}{p_2}(1 - e^{-\Lambda p_2 \tau}) - c_1)}{c_3}\right]\right). \tag{5.3}$$

6. NUMERICAL EXAMPLES

We now illustrate our results with some numerical examples, using as data the failure data set DS1 discussed by Goel (1980). This data gives rise to estimates of $N = 1348$ and $\Lambda = 0.124$ for the parameters in the Jelinski Moranda model. We also use the same cost parameters of $c_1 = 1, c_2 = 5, c_3 = 100$ suggested by Okumoto and Goel (1980) (and used by Boland and Singh (2000a) and (2000b)) in order to make comparisons with other studies.

Table 2 gives the optimal testing time T^* for different values of p (the probability of perfect fault repair), for the cases where the fixed life cycle time is $t_0 = 100$ and for the situation when we are interested in the software functioning well during a mission time of $\tau = 2, 5, 10, 20, 50, 100$. Figure 1 illustrates the behavior of T^* as a function of p plotted for a number of values of τ. Table 2 illustrates the general result (which is clear from equation (3.5)) that for a fixed life cycle time, the optimal T^* is a decreasing function of p. In other words, as the probability of a perfect repair increases, the less one needs to test the software. Note however that for a mission time of τ, the optimal time T^* is not necessarily a decreasing function of p, however for a fixed value of p it is (in general) an increasing function of τ. Table 3 gives the corresponding minimum expected costs for these optimal release times. Note that these minimal costs decrease with p and increase with τ, results which support one's intuition.

Next we consider the situation where the probability of perfect repair before (p_1) and after (p_2) release of the software may vary. As in the previous examples, we have taken $c_1 = 1, c_2 = 5, c_3 = 100, N = 1348$ and $\Lambda = 0.124$. Table 4 gives the optimal T^* when the life-cycle time $t_0 = 100$, and Table 5 gives the corresponding minimal expected costs. Note that for fixed p_2, T^* is not necessarily (as might be expected) a decreasing function of p_1. Table 4 does however suggest that for fixed p_1, T^* is a decreasing function of p_2.

Tables 6 and 7 are the optimal T^* and minimal expected costs for the situation where one is interested in a fixed mission time $\tau = 100$. As Table 6 suggests, it can be shown that in general T^* is a decreasing function of p_2.

REFERENCES

Boland, P.J. and Singh, H. (2000a). A Birth Process approach to Morand's Geometric software Reliability Model, under revision for *IEEE Transactions in Reliability*.

Boland, P.J. and Singh, H. (2000b). Determining the Optimal Release Time for Software in the Geometric Poisson Reliability Model, submitted for publication.

Brown M. and Proschan, F. (1983). Imperfect Repair. *Journal of Applied Probability*, **20**, 851-862.

Dalal, S. R. and Mallows, C. L. (1988). When to stop testing software. *Journal of American Statistical Association*, **83**, 872-879.

Farr, W. (1996). *Software Reliability Modeling Survey*, Chapter 3 in Handbook of Software Reliability Engineering, edited by M. R. Lyu, McGraw Hill.

Finkelstein M.S.(1997). Imperfect repair models for systems subject to shocks. *Applied Stochastic Models and Data Analysis*, **13**, 385-390.

Goel, A.L. (1980). Software Error Detection model with applications. *The Journal of Systems and Software*, **1**, 243-249.

Goel, A.L. and Okumoto, K. (1978). An Analysis of Recurrent Software Failures on a Real-Time Control System. *In Proceedings of the ACM Annual Technical Conference*, 496-500.

Jelinski, Z. and Moranda, P. (1972). Software Reliability Research. *Statistical Computer Performance Evaluation*, Ed. W. Freiberger, 465-84, New York, Academic.

Lyu, M. R. (1996). *Handbook of Software Reliability Engineering*, McGraw Hill.

Lim J.H. and Park D.H. (1999). Evaluation of average maintenance cost for imperfect-repair model. *IEEE Transactions in Reliability*, **48(2)**, 199-204.

Makis, V. and Jardine, A.K.S. (1992). Optimal Replacement for a general-model with Imperfect Repair. *Journal of the Operational Research Society*, **43(2)**, 111-120.

Mazzuchi, T.A. and Soyer, R. (1988). A Bayes Empirical Bayes for Software Reliability. *IEEE Transactions in Reliability*, **37**, 248-254.

McDaid, K. (1998). Deciding How Long to Test Software. Unpublished thesis, Department of Statistics, Trinity College Dublin.

Moranda, P.B. (1975). Prediction of Software Reliability Software During Debugging. *Proceedings on the 1975 Annual Reliability and Maintainability Symposium*, 327-32.

Musa, J. D., Iannino, A. and Okumoto, K. (1987). *Software Reliability: Measurement, Prediction, Application*, New York Wiley.

Okumoto, K. and Goel, A.L. (1980). Optimal Release Time for Software Systems Based on Reliability and Cost Criteria. *The Journal of Systems and Software*, **1**, 315-318.

Sheu S.H. (1998). A generalized age and block replacement of a system subject to shocks. *European Journal of Operational Research*, **108**, 345-362.

Singpurwalla, N. D. and Wilson, S. P. (1999). *Statistical Methods in Software Engineering: Reliability and Risk.* Springer.

Singpurwalla, N. D. (1991). Determining an optimal time interval for testing and debugging software. *IEEE Transactions on Software Engineering*, **17**, 313-319.

Wang H.Z. and Pham H. (1996). Optimal maintenance policies for several imperfect repair models. *International Journal of Systems Science*, **27(6)**, 543-549.

Xie, M. (2000) Software Reliability Models - Past, Present and Future. in *Recent Advances in Reliability Theory - Methodology, Practive and Inference.* Eds Limnios, N and Nikulin M., Birkhauser, 325-340.

BIOGRAPHY OF AUTHOR

Philip J. Boland is the Professor of Statistics at the National University of Ireland - Dublin (UCD). He served as Head of the Department of Statistics at UCD (University College Dublin) for the period 1986-2001. Dr. Boland has a B.A. degree from LeMoyne College (New York), a Ph.D. degree from the University of Rochester (New York), and a D.Sc. from the National University of Ireland. He is the founder and Director of the UCD Actuarial and Financial Studies degree programme at UCD. Professor Boland is an elected member of the International Statistical Institute, and a member of the Statistical and Social Inquiry Society of Ireland, the American Statistical Society, the Society of Actuaries in Ireland, the Irish Statistical Association and the Irish Mathematical Society. In 1996, he was the first person to be elected as an Honorary Member of the Society of Actuaries in Ireland, and in 2000 became the first Honorary Fellow of the Society. In 1997/1998 he was elected as the first President of the Irish Statistical Association. Over the past 20 years the main area of research of Dr. Boland has been in reliability theory, but he also has interests in statistical education, actuarial statistics, stochastic processes, the history of statistics, mathematical statistics and random numbers. He has published over 80 papers in a wide variety of journals and conference proceedings, and presently acts as an Associate Editor for "Statistics and Probability Letters" and "Lifetime Data Analysis".

Table 1. Distribution of Number of Remaining Faults at Time T

Distribution π for N	Distribution for $N - R(T)$
Poisson (θ)	Poisson $(\theta e^{-\Lambda pT})$
NegBin (κ, θ)	NegBin $(\kappa, 1/(1 + \frac{1-\theta}{\theta} e^{-\Lambda pT}))$
Bin (M, θ)	Bin $(M, \theta e^{-\Lambda pT})$

Table 2. Optimal Testing Time, T^*

p	$t_0 = 100$	$\tau = 2$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$
0.2	76.61	0	0	0	18.87	58.51	72.16
0.4	38.31	0	0	9.43	25.75	36.08	38.13
0.6	25.54	0	0	13.42	21.08	25.12	25.53
0.8	19.15	0	4.72	12.88	17.25	19.06	19.15
1.0	15.32	0	6.32	11.70	14.43	15.30	15.32

Table 3. Minimum Expected Cost

p	$t_0 = 100$	$\tau = 2$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$
0.2	15612	1631	3930	7402	12659	16623	17988
0.4	9099	1591	3701	6329	7961	8994	9199
0.6	6138	1553	3490	4933	5699	6103	6144
0.8	4608	1516	3165	3981	4418	4600	4608
1.0	3687	1480	2787	3325	3598	3685	3687

Table 4. Optimal Testing Time T^* for a given life-cycle time $t_0 = 100$ when p varies

p_1	$p_2 = 0.2$	$p_2 = 0.4$	$p_2 = 0.6$	$p_2 = 0.8$	$p_2 = 1.0$
0.2	76.61	40.42	4.53	0	0
0.4	51.02	38.31	27.51	18.54	10.36
0.6	40.01	31.89	25.54	20.50	16.22
0.8	33.21	27.18	22.66	19.15	16.25
1.0	28.54	23.72	20.20	17.52	15.32

Table 5. Minimum Expected Cost (for $t_0 = 100$) when p varies

p_1	$p_2 = 0.2$	$p_2 = 0.4$	$p_2 = 0.6$	$p_2 = 0.8$	$p_2 = 1.0$
0.2	15612	14170	11201	8425	6740
0.4	10090	9099	8117	7238	6422
0.6	7463	6744	6138	5640	5213
0.8	5955	5395	4956	4608	4318
1.0	4975	4518	4173	3906	3687

Table 6. Optimal Testing Time T^* for a fixed mission time $\tau = 100$ when p varies

p_1	$p_2 = 0.2$	$p_2 = 0.4$	$p_2 = 0.6$	$p_2 = 0.8$	$p_2 = 1.0$
0.2	72.16	36.59	4.31	0	0
0.4	52.69	38.13	27.42	18.53	10.36
0.6	41.11	31.95	25.53	20.50	16.22
0.8	33.93	27.25	22.66	19.15	16.25
1.0	29.04	23.78	20.21	17.52	15.32

Table 7. Minimum Expected Cost (for $\tau = 100$) when p varies

p_1	$p_2 = 0.2$	$p_2 = 0.4$	$p_2 = 0.6$	$p_2 = 0.8$	$p_2 = 1.0$
0.2	17988	14431	11203	8425	6740
0.4	10655	9199	8128	7239	6422
0.6	7702	6786	6144	5641	5213
0.8	6086	5418	4959	4608	4318
1.0	5058	4532	4175	3906	3687

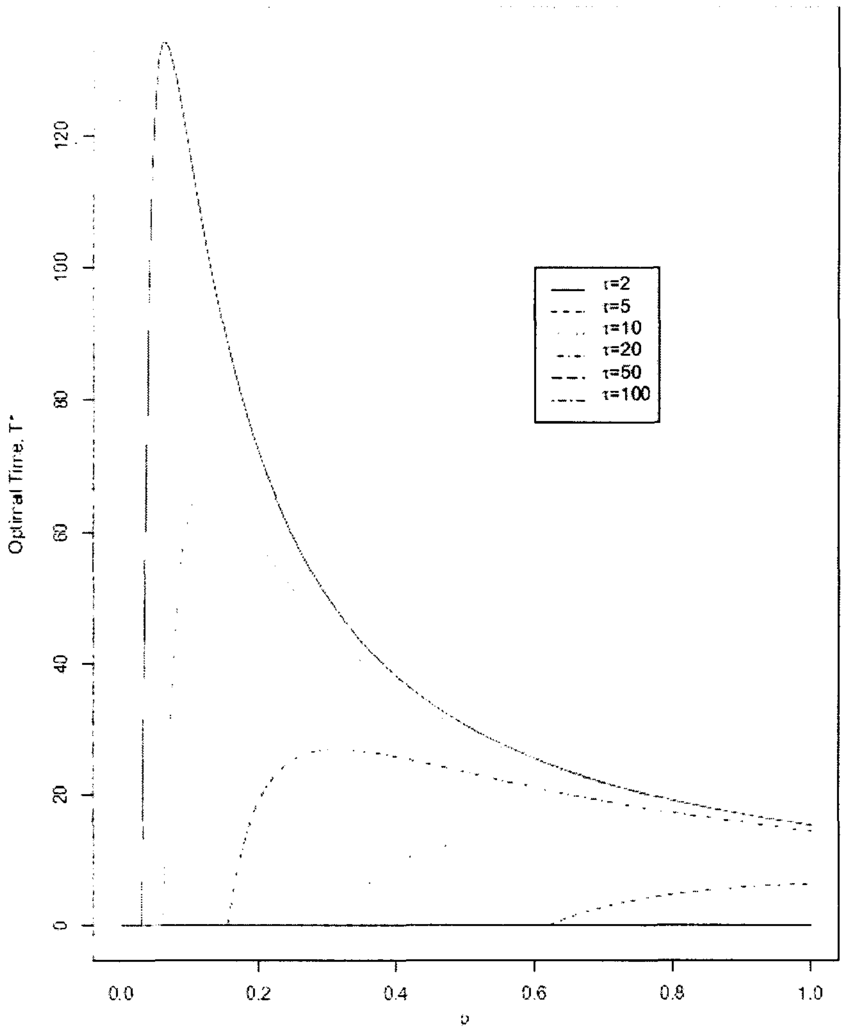

Figure 1. Optimal Time, T^*, for Various Values of τ