# Optimal Release Times for Software Systems with Scheduled Delivery Time Based on the HGDM

Rong–Huei Hou, Sy–Yen Kuo, *Member*, *IEEE*,
and Yi–Ping Chang

**Abstract**—The Hyper-Geometric Distribution software reliability growth Model (HGDM) was developed to estimate the number of remaining software faults after completing the test/debug phase. An important problem in the software development process is to determine when to stop testing and release the software to the users. In this paper, the cost optimal release policy, which minimizes the total expected software cost, is discussed. The total expected software cost here includes the penalty cost, which should be paid by the manufacturer if the software is delivered after the scheduled delivery time. The underlying software reliability growth model in our approach is the HGDM. Numerical examples are presented for illustration.

**Index Terms**—Software reliability, optimum software release time, software testing, cost-benefit analysis, software cost model.

## 1 Introduction

In recent years, computer systems have been widely applied to the control of many complex and critical systems. The breakdown of a computer system, caused by software faults, may result in tremendous damage for social life. Therefore, software reliability is one of the key issues in the software product development. In the literature, many Software Reliability Growth Models (SRGMs) have been developed [1], [2], [3], [4], [5], [6]. The SRGMs are usually used to estimate the number of remaining faults, software reliability, and other assessment measures.

In addition to fulfilling the requirement on software reliability, the software project manager would like to know when to stop testing and release the software to the users. In general, the longer the software is tested, the more reliable it tends to be. But as for any other product, it is important to release the software product as early as possible. Therefore, it is desirable to determine the optimal release time [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Such a decision problem is called an optimal software release problem. Okumoto and Goel [9] discussed the *cost* optimal software release policy which minimizes the total expected software cost. However, the cost model they proposed excludes the penalty cost due to the delay from the scheduled delivery time. In fact, the delay by the test will incur additional cost and the late release for operational use will also lead to users' dissatisfaction. Hence, it is usually assumed that additional penalty cost should be paid if a software is released after the scheduled delivery time. Therefore, Koch and Kubat [10] and Yamada et al. [11] discussed the cost optimal release policy with scheduled software delivery time. The underlying SRGMs in their approach are based on the Non-Homogeneous Poisson Process (NHPP).

The Hyper-Geometric Distribution software reliability growth Model (HGDM), first proposed by Tohma et al. [6], was shown to be attractive. Tohma et al. [17], [18], Jacoby and Tohma. [19], [20], and Hou et al. [21], [22], [23] have made a series of studies on the HGDM recently. In this paper, we investigate the cost optimal release policy for software systems with scheduled delivery time under the HGDM with exponential or logistic learning factor [21]. The overall organization of this paper is as follows. A brief review of the HGDM with exponential or logistic learning factor is given in Section 2. The software cost model including the penalty cost is proposed in Section 3. The cost optimal software release policy with scheduled delivery time based on the HGDM is discussed in Section 4. Numerical results are presented for illustration in Section 5 followed by the conclusions in Section 6.

- *R.-H. Hou and S.-Y. Kuo are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC. E-mail: sykuo@cc.ee.ntu.edu.tw.*
- *Y.-P. Chang is with the Department of Business Mathematics, Soochow University, Taipei, Taiwan, Republic of China.*

*Manuscript received March 1, 1995; revised November 1, 1995.*
*For information on obtaining reprints of this article, please send e-mail to: transcom@computer.org, and reference IEEECS Log Number C96303.*

## 2 Review of HGDM

In this section, we briefly review the Hyper-Geometric Distribution software reliability growth Model (HGDM) [17], [18], [19], [20], [21]. The HGDM has been developed to estimate the number of remaining software faults after the test/debug phase. In general, a program is assumed to have $m$ faults initially when the test/debug phase starts. The collection of test operations performed in a day or a week is called a *test instance*. The test/debug phase consists of consecutive applications of test instances. Test instances are denoted by $t_i$, $i = 1, 2, \ldots, n$ in accordance with the order of applying them. The *sensitivity factor*, $w_i$, represents how many faults are discovered during the application of test instance $t_i$. Each fault is classified into one of two categories: **newly discovered** faults or **rediscovered** faults. Some of the faults detected by $t_i$ may have been detected previously by the application of $t_1$ through $t_{i-1}$ test instances. Therefore, the number of newly detected faults during the application of the $i$th test instance is not necessarily equal to $w_i$.

Considering the application of $t_i$, let $C_{i-1}$ be the number of faults already detected so far by $t_1, t_2, \ldots, t_{i-1}$ and $N_i$ be the number of faults newly detected by $t_i$. Then, some of the $w_i$ faults may be those that are already counted in $C_{i-1}$, and the remaining $w_i$ faults account for the newly detected faults. With the assumption that new faults will not be inserted into the program while correction is being performed, the conditional probability $Prob(N_i = x_i \mid m, w_i, C_{i-1})$ can be formulated as

$$Prob\left(N_i = x_i \middle| m, w_i, C_{i-1}\right) = \frac{\binom{m - C_{i-1}}{x_i}\binom{C_{i-1}}{w_i - x_i}}{\binom{m}{w_i}}, \tag{1}$$

where $\max(0,\ w_i - C_{i-1}) \le x_i \le \min(w_i, m - C_{i-1})$ for all $i \ge 0$, $C_{i-1} = \sum_{k=1}^{i-1} x_k$, $C_0 = 0$, $x_0 = 0$, and $x_k$ is an observed instance of $N_k$. The expected value of $C_i$ denoted by $EC_i$ is [17], [18]. [19], [20]

$$\begin{cases} EC_0 = 0, \\ EC_i = m\left[1 - \Pi_{j=1}^{i}\left(1 - p_j\right)\right], \quad i = 1, 2, \ldots, \end{cases} \tag{2}$$

where

$$p_i = w_i / m. \tag{3}$$

There are various functions for $p_i$ presented in [17], [18], [19], [20]. Hou et al. [21] have proposed two $p_i$ functions based on the exponential and the S-shaped learning curves, respectively. The

first $p_i$ function is

$$p_i = p_{LT}(1 - e^{-ai}), a > 0,\ 0 < p_{LT} \le 1, \tag{4}$$

which is called the *exponential learning factor*. The second $p_i$ function is

$$p_i = p_{LT}\frac{1}{1 + be^{-ai}}, a > 0,\ b > 0,\ 0 < p_{LT} \le 1, \tag{5}$$

which is called the *logistic learning factor*.

In previous research [17], [18], [19], [20], [21], although $p_i$ is assumed to be a smooth and deterministic function of $i$, the HGDMs present adequately the stochastic structure of $x_1, x_2, \ldots, x_n$ newly discovered faults at the successive test instances. In some applications, the HGDMs can fit observed data quite well.

In the following sections, we will discuss the cost optimal software release policy with scheduled software delivery time. The underlying software reliability growth model is the HGDM with exponential or logistic learning factor.

## 3 Software Cost Model

Okumoto and Goel [9] discussed the software optimal release policy from the cost-benefit viewpoint. However, the cost model they proposed excludes the penalty cost due to the delay from the scheduled delivery time. In general, additional penalty cost should be paid if a software is released after the scheduled delivery time [10], [11]. In this paper, the penalty cost function for the HGDM is

$$C_D(i) = \begin{cases} 0, & i < D; \\ c_4 + c_5 g(i - D), & i \ge D; \end{cases} \tag{6}$$

where

$c_4, c_5$ = nonnegative real numbers,
$D$ = scheduled software delivery time ($D > 0$),
$C_D(i)$ = penalty cost function due to the delay from the scheduled software release time.

In addition, the longer the software is delayed delivering, the more the manufacturer should pay the penalty cost. Thus, $g(i)$ is assumed to satisfy the following three conditions A1–A3.

A1: $g(0) = 0$;
A2: $g(i)$ is increasing in $i$;
A3: $g(i)$ is a convex function of $i$. That is, $g(i + 1) + g(i - 1) \ge 2g(i)$ for all $i \ge 1$.

For example, $g(i)$ can be one of the following:

1) $g(i) = i$;
2) $g(i) = i^h$, $h > 1$;
3) $g(i) = e^{hi} - 1$, $h > 0$.

Including the penalty cost, the total expected software cost for the HGDM is given by

$$CT(i) = c_1 EC_i + c_2(m - EC_i) + c_3 i + C_D(i),\ i = 0, 1, 2, \ldots, \tag{7}$$

where

$CT(i)$ = total expected cost when the software is released at the $i$th test instance,
$c_1$ = cost of fixing a fault during the testing phase,
$c_2$ = cost of fixing a fault during the operational phase ($c_2 > c_1 > 0$),
$c_3$ = cost of per unit time of software testing ($c_3 > 0$).

## 4 Cost Optimal Release Policy with Scheduled Delivery Time

Since $CT(i)$ is the evaluation criterion, the cost optimal release problem is to find a cost optimal release time $I^*$ which minimizes (7). From (2), we have

$$\begin{cases} CT(0) = c_2 m, \\ CT(i) = c_1 m + (c_2 - c_1)m\Pi_{j=1}^{i}(1 - p_j) + c_3 i \\ \qquad\qquad + C_D(i),\ \ i = 1, 2, \ldots . \end{cases} \tag{8}$$

For convenience, let

$$C^*(i) = \frac{c_3 + C_D(i) - C_D(i-1)}{(c_2 - c_1)m},\ \ i = 1, 2, \ldots \tag{9}$$

and

$$\begin{cases} \delta(1) = p_1; \\ \delta(i) = p_i\Pi_{j=1}^{i-1}(1 - p_j),\ \ i = 2, 3, \ldots . \end{cases} \tag{10}$$

Since

$$\begin{aligned} CT(1) - CT(0) &= c_3 - (c_2 - c_1)mp_1 + C_D(1) \\ &= (c_2 - c_1)m\{C^*(1) - \delta(1)\} \end{aligned} \tag{11}$$

and

$$\begin{aligned} CT(i+1) - CT(i) &= c_3 - (c_2 - c_1)mp_{i+1}\prod_{j=1}^{i}(1 - p_j) + C_D(i+1) - C_D(i) \\ &= (c_2 - c_1)m\{C^*(i+1) - \delta(i+1)\},\ \ i = 1, 2, \ldots, \end{aligned}$$

the following lemma can be easily obtained.

LEMMA 1. *Assume $c_2 > c_1 > 0$ and $c_3 > 0$. For $i = 0, 1, 2, \ldots$, we have*:

1) *if $C^*(i + 1) > \delta(i + 1)$, then $CT(i + 1) > CT(i)$;*
2) *if $C^*(i + 1) < \delta(i + 1)$, then $CT(i + 1) < CT(i)$;*
3) *if $C^*(i + 1) = \delta(i + 1)$, then $CT(i + 1) = CT(i)$.*

Let

$$\Delta(i) = p_{i+1} - p_i - p_{i+1}p_i,\ i = 1, 2, \ldots, \tag{12}$$

and then

$$\begin{cases} \delta(2) - \delta(1) = \Delta(1); \\ \delta(i+1) - \delta(i) = \{\Pi_{j=1}^{i-1}(1 - p_j)\}\Delta(i),\ \ i = 2, 3, \ldots . \end{cases} \tag{13}$$

Define

$$I = inf\{i \ge 1 : \Delta(i) < 0\}. \tag{14}$$

For the exponential and the logistic learning factors, we have the following lemmas, respectively.

LEMMA 2. *If $p_i = p_{LT}(1 - e^{-ai})$, then*:

1) *$\delta(i) \le \delta(i + 1)$ for $1 \le i \le I - 1$;*
2) *$\delta(i) > \delta(i + 1)$ for $i \ge I$.*

PROOF. If $p_i = p_{LT}(1 - e^{-ai})$, then $\Delta(i)$ is decreasing in $i$ [21] and $\lim_{i\to\infty}\Delta(i) = -p_{LT}^2 < 0$. Hence, from the definition of $I$, $I$ is finite and unique. Furthermore, from (13), this lemma can be proved.

LEMMA 3. *If $p_i = p_{LT}/(1 + be^{-ai})$, then*:

1) *$\Delta(i)$ is decreasing for $i \ge I$;*
2) *$\delta(i) \le \delta(i + 1)$ for $1 \le i \le I - 1$ and $\delta(i) > \delta(i + 1)$ for $i \ge I$.*

PROOF. From (12), we have

$$\begin{aligned} \Delta(i) &= \frac{p_{LT}}{1 + be^{-a(i+1)}} - \frac{p_{LT}}{1 + be^{-ai}} - \frac{p_{LT}^2}{\left(1 + be^{-a(i+1)}\right)\left(1 + be^{-ai}\right)} \\ &= p_{LT}\frac{e^{-a(i+1)}}{\left(1 + be^{-a(i+1)}\right)\left(1 + be^{-ai}\right)}(be^{a} - b - p_{LT}e^{a(i+1)}). \end{aligned} \tag{15}$$

Therefore,

$$\Delta(i) < 0 \text{ if and only if } be^{a} - b - p_{LT}e^{a(i+1)} < 0. \tag{16}$$

In addition,

$$\Delta(i+1) - \Delta(i) = p_{LT}\frac{H(i)}{\left(1+be^{-a(i+2)}\right)\left(1+be^{-a(i+1)}\right)\left(1+be^{-ai}\right)}, \quad (17)$$

where

$$H(i) = (e^{-a}-1)be^{-2a(i+1)}(b - be^{a} + (p_{LT}-1)e^{a(i+1)} + (p_{LT}+1)e^{a(i+2)}).$$

Since $a > 0$, i.e., $e^{-a} - 1 < 0$, we have

$H(i) < 0$ if and only if $b - be^{a} + (p_{LT}-1)e^{a(i+1)} + (p_{LT}+1)e^{a(i+2)} > 0$. (18)

Assuming $\Delta(i') < 0$, from (16) and (18), we have $H(i') < 0$ owing to

$$b - be^{a} + (p_{LT}-1)e^{a(i'+1)} + (p_{LT}+1)e^{a(i'+2)}$$
$$= -\left(be^{a} - b - p_{LT}e^{a(i'+1)}\right) + e^{a(i'+1)}\left((p_{LT}+1)e^{a} - 1\right) > 0.$$

That is, if $\Delta(i') < 0$, then $\Delta(i'+1) < \Delta(i')$. Therefore, from the definition of $I$, $\Delta(i)$ is decreasing for $i \geq I$. Furthermore, from (13), we have $\delta(i) \leq \delta(i+1)$ for $1 \leq i \leq I-1$ and $\delta(i) > \delta(i+1)$ for $i \geq I$. □

For convenience, we define

$$I_f = inf\{i \geq I : \delta(i+1) \leq C^*(i+1)\}, \quad (19)$$

and

$$S = \{D-1 \leq i \leq I-1 : \delta(i) > C^*(i) \text{ and } \delta(i+1) \leq C^*(i+1)\} \quad (20)$$

LEMMA 4. *Suppose that $g(i)$ satisfies conditions* A1–A3. *If $p_i = p_{LT}(1 - e^{-ai})$ or $p_i = p_{LT}/(1 + be^{-ai})$, then $I_f$ is finite and unique.*

PROOF. From the definition of $\delta(i)$, it is true that $\delta(i)$ is decreasing in $i$ and $\lim_{i\to\infty}\delta(i) = 0$. From the definition of $C^*(i)$, we have $C^*(i) \geq c_3/[(c_2 - c_1)m] > 0$ for $i \geq 1$, and then $C^*(i+1) > 0$ for all $i \geq I$. Based on the above facts, this lemma is proved. □

In the following, the theorem on the cost optimal release time $I^*$ is presented.

THEOREM 1. *Suppose that $g(i)$ satisfies conditions* A1–A3. *If $p_i = p_{LT}(1 - e^{-ai})$ or $p_i = p_{LT}/(1 + be^{-ai})$,* we have:

1) *if $D > I$,* then $I^*$ *satisfies* $CT(I^*) = \min_{i\in\{0, I_f\}} CT(i)$;
2) *if $D \leq I$, then $I^*$ satisfies $CT(I^*) = \min_{i\in S\cup\{0,I_f\}} CT(i)$, where $I_f$ and $S$ are defined in* (19) *and* (20).

PROOF. In the following proof, we only consider the case $p_i = p_{LT}(1 - e^{-ai})$. However, the argument is the same for the case $p_i = p_{LT}/(1 + be^{-ai})$, except that Lemma 3 will be referenced instead of Lemma 2.

If $p_i = p_{LT}(1 - e^{-ai})$, from (6) and (9), we have

$$C^*(1) = C^*(2) = \ldots = C^*(D-1) \leq C^*(D) \leq C^*(D+1) \leq \ldots. \quad (21)$$

If $D > I$ and $C^*(I) \geq \delta(I)$, by Lemma 2, we have $C^*(i) > \delta(i)$ for $i = 1, 2, \ldots, I-1, I+1, \ldots$. By Lemma 1, we have $CT(0) < CT(1) < \ldots < CT(I+1) \leq CT(I) < CT(I+1) < \ldots$. That is, the cost is minimum at $i = 0$. Therefore, $I^* = 0$.

If $D > I$ and $\delta(I) > C^*(I) \geq \delta(1)$, by Lemma 2 and (21), there exists two finite and unique integers $I_A$ ($I_A < I$) and $I_f$ (defined in (19)) such that

$$C^*(i) \geq \delta(i) \text{ for } i = 1, 2, \ldots, I_A;$$
$$C^*(i) < \delta(i) \text{ for } i = I_A+1, I_A+2, \ldots, I_f;$$
$$C^*(i) \geq \delta(i) \text{ for } i = I_f+1, I_f+2, \ldots.$$

Therefore, by Lemma 1, we have

$$CT(0) \leq CT(1) \leq \ldots \leq CT(I_A);$$
$$CT(I_A) > CT(I_A+1) > \ldots > CT(I_f);$$
$$CT(I_f) \leq CT(I_f+1) \leq \ldots.$$

This means if $CT(0) \leq CT(I_f)$, then $I^* = 0$; otherwise, $I^* = I_f$.

If $D > I$ and $C^*(I) < \delta(1)$, by Lemma 2 and (21), we have

$$C^*(i) < \delta(i) \text{ for } i = 1, 2, \ldots, I_f;$$
$$C^*(i) \geq \delta(i) \text{ for } i = I_f+1, I_f+2, \ldots.$$

Therefore, by Lemma 1, we have

$$CT(0) > CT(1) > \ldots > CT(I_f);$$
$$CT(I_f) \leq CT(I_f+1) \leq \ldots.$$

Hence, $I^* = I_f$. From the above arguments and simple arrangement, condition 1 of Theorem 1 holds.

If $D \leq I$, there may exist some integers $k$s $\in \{D-1, D, \ldots, I-1\}$ such that

$$\delta(k) > C^*(k) \text{ and } \delta(k+1) \leq C^*(k+1).$$

By Lemma 1, we have

$$CT(k-1) > CT(k) \text{ and } CT(k+1) \geq CT(k).$$

Therefore, we have $I^* \in S$ and $S$ is defined in (20).

Applying the same argument in the case of $D > I$, the remaining results for the case of $D \leq I$ can be obtained. Moreover, by simple arrangement, condition 2 of Theorem 1 holds. □

Note that Theorem 1 can be consolidated into $I^* \in S \cup \{0, I_f\}$. From the definitions of $I_f$ and $S$, we have $I_f = \max\{S \cup \{0, I_f\}$. Therefore, the cost optimal release time is $I^* \leq I_f$.

In some applications, the penalty cost due to the delay from the scheduled delivery time may be very small and can be negligible. In this case, we assume $c_4 = c_5 = 0$, and then the following corollary can be easily obtained by applying Theorem 1.

COROLLARY 1. *Assume $c_4 = 0$ and $c_5 = 0$. If $p_i = p_{LT}(1 - e^{-ai})$ or $p_i = p_{LT}/(1 + be^{-ai})$, then $I^* \in \{0, I_f\}$. That is, if $CT(I_f) \leq CT(0)$ then $I^* = I_f$; otherwise, $I^* = 0$.*

In Theorem 1, we have also proposed the procedure of determining the cost optimal release time $I^*$ based on the HGDM with exponential or logistic learning factor, and $I^*$ can be determined and shown to be finite. In addition, a special case of $C_D(i)$ is discussed in the above corollary.

## 5 NUMERICAL EXAMPLES

In this section, numerical examples are used to illustrate the cost optimal release policy with scheduled delivery time based on the HGDM with exponential or logistic learning factor. The data used in this analysis is the test/debug data of a software system [24]. Since the test data is reported per week, a test instance is "a week of observation." The data is the collection of the cumulative number of faults for 24 test instances. The parameters of $CT(i)$ are taken from [9]: $c_1$ = 1$ per fault, $c_2$ = 5$ per fault, $c_3$ = 10$ per week, and $c_4$ = 1$ . The various values of $c_5$ in the penalty cost function $C_D(i)$ are assumed to be 1, 5, 10, 20, and 40. The following three cases of $g(i)$ are considered:

1) $g(i) = i$;
2) $g(i) = i^2$;
3) $g(i) = e^i - 1$.

### 5.1 Exponential Learning Factor

The least squares estimates of the parameters of the HGDM with exponential learning factor are $\hat{m}$ = 2300.8, $\hat{a}$ = 0.1206, and $\hat{p}_{LT}$ =0.1602 [21]. Tables 1, 2, and 3 show the relationship of the cost optimal release time $I^*$, the total expected software cost $CT(I^*)$, the parameter $c_5$, and the scheduled delivery time $D$. Function $g(i)$ is assumed to be $g(i) = i$, $g(i) = i_2$, and $g(i) = e^i - 1$ for Tables 1, 2, and 3, respectively. Fig. 1 shows the dependence of the total expected cost $CT(i)$ on $c_5$ with $D = 10$ and $g(i) = i^2$.

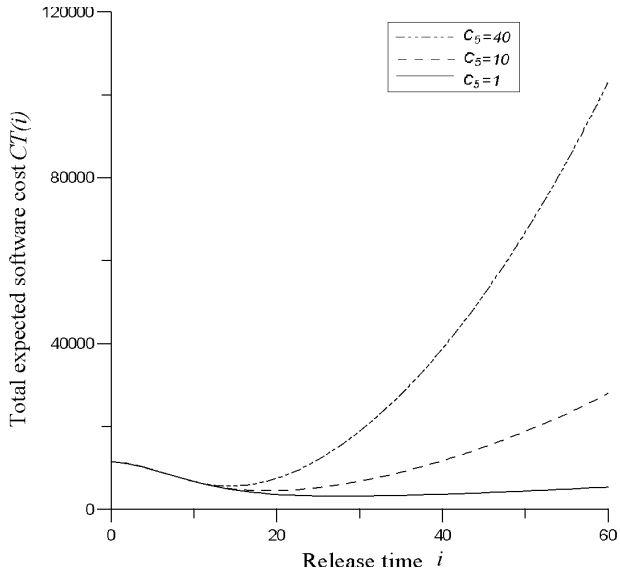


Fig. 1. Dependence of the total expected cost $CT(i)$ on $c_5$ based on the HGDM with exponential learning factor ($D = 10$ and $g(i) = i^2$).

Considering the case with $c_5 = 5$, $D = 10$, and $g(i) = i^2$, we have $I^* = 22$. Since 24 test instances have been collected, test engineers should stop testing and release the software immediately. Considering another case with $c_5 = 5$, $D = 20$, and $g(i) = i^2$, we have $I^* = 26$. Hence, the 25th and 26th test instances should be collected and tested by the test engineers before the software is released.

If the penalty cost is not considered (i.e., $c_4 = c_5 = 0$), we have the cost optimal release time $I^* = 37$ (applying Corollary 1). With $c_5 = 1$ and $g(i) = i^2$, Table 2 shows $I^* = 28$ and $I^* = 30$ for $D = 10$ and $D = 20$, respectively. Obviously, the effect of the penalty cost is significant, especially for the case $g(i) = e^i - 1$. Hence, earlier release of the software system is a cost-effective policy if the penalty cost is considered.

### 5.2 Logistic Learning Factor

The least squares estimates of the parameters of the HGDM with logistic learning factor are $\hat{m}$ = 2313.4, $\hat{a}$ =0.3362, $\hat{b}$ = 5.5395, and $\hat{p}_{LT}$ = 0.1363 [21]. Tables 4, 5, and 6 show the relationship of the cost optimal release time $I^*$, the total expected software cost $CT(I^*)$, the parameter $c_5$, and the scheduled delivery time $D$. Function $g(i)$ is assumed to be $g(i) = i$, $g(i) = i^2$, and $g(i) = e^i - 1$ for Tables 4, 5, and 6, respectively. Fig. 2 shows the dependence of the total expected cost $CT(i)$ on $c_5$ with $D = 10$ and $g(i) = i^2$.

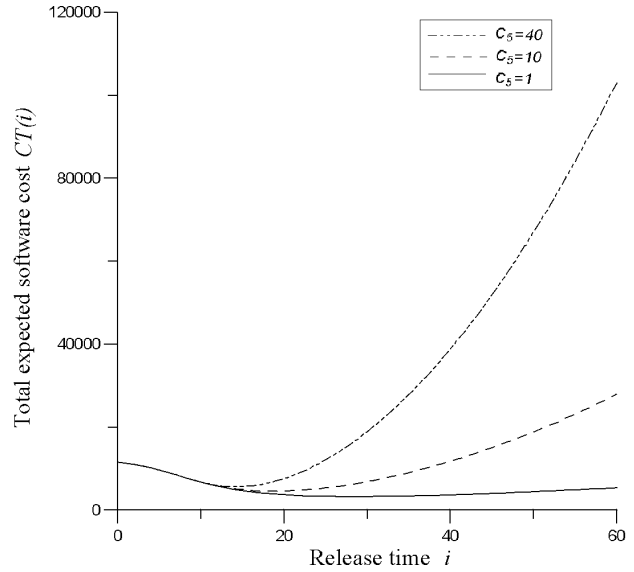


Fig. 2. Dependence of the total expected cost $CT(i)$ on $c_5$ based on the HGDM with logistic learning factor ($D = 10$ and $g(i) = i^2$).

From Tables 1, 2, 3, 4, 5, 6, there are some facts observed as follows.

1) As $c_5$ increases, the total expected cost $CT(I^*)$ increases but the cost optimal release time $I^*$ decreases.
2) As the slope of $g(i)$ increases, the total expected cost $CT(I^*)$ increases but the cost optimal release time $I^*$ decreases.
3) Earlier release of the software system is a cost-effective policy if the penalty cost is included.
4) The cost optimal release times for the exponential learning factor and the logistic learning factor are nearly equal. That is, there is insignificant difference between the exponential learning factor and the logistic learning factor in determining the optimal release time.

## 6 Conclusions

In this paper, we have discussed the cost optimal release policy with scheduled delivery time based on the HGDM with exponential or logistic learning factor. The effect of the penalty cost on the cost optimal release policy was analyzed. It is concluded that earlier release of the software system is a cost-effective policy if the penalty cost is considered. In addition, the cost optimal release time $I^*$ has been shown to be finite.

## Acknowledgment

We would like to express our gratitude for the support of the National Science Council, Taiwan, R.O.C., under Grants NSC84–2213–E002–035. Reviewers' comments are also highly appreciated.

TABLE 1
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $I^*$, $CT(I^*)$, $c_5$, AND D BASED ON THE HGDM WITH EXPONENTIAL LEARNING FACTOR ($g(i) = i$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| $c_5$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ |
| 1 | 37 | 2757.75 | 37 | 2747.75 |
| 5 | 35 | 2859.99 | 35 | 2809.99 |
| 10 | 33 | 2979.06 | 33 | 2879.06 |
| 20 | 31 | 3196.81 | 31 | 2996.81 |
| 40 | 28 | 3576.15 | 28 | 3176.15 |

TABLE 2
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $I^*$, $CT(I^*)$, $c_5$, AND $D$ BASED ON THE HGDM WITH EXPONENTIAL LEARNING FACTOR ($g(i) = i^2$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| c5 | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ |
| 1 | 28 | 3180.15 | 30 | 2897.40 |
| 5 | 22 | 3982.97 | 26 | 3125.46 |
| 10 | 19 | 4497.71 | 24 | 3236.33 |
| 20 | 16 | 5074.09 | 23 | 3341.65 |
| 40 | 14 | 5613.46 | 21 | 3422.65 |

TABLE 3
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $I^*$, $CT(I^*)$, $c_5$, AND $D$ BASED ON THE HGDM WITH EXPONENTIAL LEARNING FACTOR ($g(i) = e^i - 1$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| $c_5$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ |
| 1 | 16 | 4756.52 | 24 | 3129.93 |
| 5 | 14 | 5241.45 | 23 | 3257.07 |
| 10 | 14 | 5509.44 | 22 | 3326.86 |
| 20 | 13 | 5725.45 | 22 | 3390.75 |
| 40 | 12 | 6012.14 | 21 | 3451.39 |

TABLE 4
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $I^*$, $CT(I^*)$, $c_5$, AND $D$ BASED ON THE HGDM WITH LOGISTIC LEARNING FACTOR ($g(i) = i$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| $c_5$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ |
| 1 | 38 | 2799.73 | 38 | 2789.73 |
| 5 | 36 | 2908.06 | 36 | 2858.06 |
| 10 | 34 | 3033.36 | 34 | 2933.36 |
| 20 | 31 | 3260.07 | 31 | 3060.07 |
| 40 | 28 | 3649.09 | 28 | 3249.09 |

TABLE 5
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $I^*$, $CT(I^*)$, $c_5$, AND $D$ BASED ON THE HGDM WITH LOGISTIC LEARNING FACTOR ($g(i) = i^2$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| $c_5$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ | Optimal release time: $I^*$ | Total expected cost: $CT(I^*)$ |
| 1 | 28 | 3253.10 | 31 | 2961.07 |
| 5 | 22 | 4059.72 | 26 | 3202.99 |
| 10 | 19 | 4561.41 | 24 | 3315.54 |
| 20 | 16 | 5117.67 | 22 | 3419.72 |
| 40 | 14 | 5648.58 | 21 | 3496.31 |

TABLE 6
RELATIONSHIP OF THE COST OPTIMAL RELEASE TIME $l^*$, $CT(l^*)$, $c_5$, AND $D$ BASED ON THE HGDM WITH LOGISTIC LEARNING FACTOR ($g(i) = e^i - 1$)

| | Scheduled delivery time: $D = 10$ | | Scheduled delivery time: $D = 20$ | |
|---|---|---|---|---|
| $c_5$ | Optimal release time: $l^*$ | Total expected cost: $CT(l^*)$ | Optimal release time: $l^*$ | Total expected cost: $CT(l^*)$ |
| 1 | 16 | 4800.10 | 24 | 3209.14 |
| 5 | 14 | 5276.57 | 23 | 3335.65 |
| 10 | 14 | 5544.56 | 22 | 3403.61 |
| 20 | 13 | 5761.29 | 22 | 3467.50 |
| 40 | 12 | 6053.60 | 21 | 3525.04 |

## REFERENCES


[1] A.L. Goel and K. Okumoto, "Time-Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures," *IEEE Trans. Reliability*, vol. 28, no. 3, pp. 206–211, Aug. 1979.

[2] J.D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," *Proc. Seventh Int'l Conf. Software Eng.*, pp. 230–238, 1984.

[3] M. Ohba, "Software Reliability Analysis Models," *IBM J. Research and Development*, vol. 28, no. 4, pp. 428–443, July 1984.

[4] S. Yamada, M. Ohba, and S. Osaki, "S–Shaped Software Reliability Growth Models and Their Applications," *IEEE Trans. Reliability*, vol. 33, no. 4, pp. 289–292, Oct. 1984.

[5] J.D. Musa, A. Iannino, and K. Okumoto, *Software Reliability—Measurement, Prediction, Application*. New York: McGraw-Hill, 1987.

[6] Y. Tohma, K. Tokunaga, S. Nagase, and Y. Murata, "Structural Approach to the Estimation of the Number of Residual Software Faults Based on the Hyper-Geometric Distribution," *IEEE Trans. Software Eng.*, vol. 15, no. 3, pp. 345–355, Mar. 1989.

[7] E.H. Forman and N.D. Singpurwalla, "An Empirical Stopping Rule for Debugging and Testing Software," *J. Amer. Statistical Assoc.*, vol. 72, pp. 750–757, 1977.

[8] E.H. Forman and N.D. Singpurwalla, "Optimal Time Intervals for Testing on Computer Software Errors," *IEEE Trans. Reliability*, vol. 28, pp. 250–253, 1979.

[9] K. Okumoto and A.L. Goel, "Optimum Release Time for Software Systems Based on Reliability and Cost Criteria," *J. System Software*, vol. 1, pp. 315–318, 1980.

[10] H.S. Koch and P. Kubat, " Optimal Release Time of Computer Software," *IEEE Trans. Software Eng.*, vol. 9, no. 3, pp. 323–327, Mar. 1983.

[11] S. Yamada, H. Narihisa, and S. Osaki, "Optimum Release Policies for a Software System with a Scheduled Delivery Time," *Int'l J. Systems Science*, vol. 15, pp. 905–914, 1984.

[12] S.M. Ross, "Software Reliability: The Stopping Problem," *IEEE Trans. Software Eng.*, vol. 11, no. 12, pp. 1,472–1,476, Dec. 1985.

[13] S. Yamada and S. Osaki, "Cost-Reliability Optimal Release Policies for Software Systems," *IEEE Trans. Reliability*, vol. 34, no. 5, pp. 422–424, May 1985.

[14] P.K. Kapur and R.B. Garg, "Cost–Reliability Optimum Release Policies for a Software System under Penalty Cost," *Int'l J. Systems Science*, vol. 20, pp. 2,547–2,562, 1989.

[15] S.R. Dalal and C.L. Mallows, "Some Graphical Aids for Deciding When to Stop Testing Software," *IEEE J. Selected Areas in Comm.*, vol. 8, no. 2, pp. 169–175, 1990.

[16] M. Xie, *Software Reliability Modeling*. Singapore: World Scientific Publisher, 1991.

[17] Y. Tohma, H. Yamano, M. Ohba, and R. Jacoby, "Parameter Estimation of the Hyper-Geometric Distribution Model for Real Test/Debug Data," *Proc. Second Int'l Symp. Software Reliability Eng.*, pp. 28–34, May 1991.

[18] Y. Tohma, H. Yamano, M. Ohba, and R. Jacoby, "The Estimation of Parameters of the Hypergeodetric Distribution and Its Application to the Software Reliability Growth Model," *IEEE Trans. Software Eng.*, vol. 17, no. 5, pp. 483–489, May 1991.

[19] R. Jacoby and Y. Tohma, "The Hyper-Geometric Distribution Software Reliability Growth Model (HGDM): Precise Formulation and Applicability," *Proc. COMPSAC 90*, pp. 13–19, 1990.

[20] R. Jacoby and Y. Tohma, "Parameter Value Computation by Least Square Method and Evaluation of Software Availability and Reliability at Service-Operation by the Hyper-Geometric Distribution Software Reliability Growth Model (HGDM)," *Proc. 13th Int'l Conf. Software Eng.*, pp. 226–237, 1991.

[21] R.H. Hou, S.Y. Kuo, and Y.P. Chang, "Applying Various Learning Curves to Hyper-Geometric Distribution Software Reliability Growth Model," *Proc. Fifth Int'l Symp. Software Reliability Eng.*, pp. 7–16, Nov. 1994.

[22] R.H. Hou, I.Y. Chen, Y.P. Chang, and S.Y. Kuo, "Optimal Release Policies for Hyper-Geometric Distribution Software Reliability Growth Model with Scheduled Delivery Time," *Proc. Asia-Pacific Software Eng. Conf.*, pp. 445–452, Dec. 1994.

[23] R.H. Hou, S.Y. Kuo, and Y.P. Chang, "Hyper-Geometric Distribution Software Reliability Growth Model with Imperfect Debugging," *Proc. Sixth Int'l Symp. Software Reliability Eng.*, pp. 195–200, Oct. 1995.

[24] A.L. Goel, "Software Reliability Modeling and Estimation Technique," final technical report RADC–TR–82–263, 1983.