数理解析研究所講究録
1079 巻 1999 年 31-40

THE SOFTWARE RELEASE GAMES REVISED

伊東陽治[†], 土肥正[†], 寺岡義伸[‡], 尾崎俊治[†]
YOUJI ITO[†], TADASHI DOHI[†], YOSHINOBU TERAOKA[‡] and SHUNJI OSAKI[†]
[†]Department of Industrial and Systems Engineering, Hiroshima University
4-1 Kagamiyama 1 Chome, Higashi-Hiroshima 739-8527, Japan.
[‡] Department of Mathematics and Information Sciences, Osaka Prefecture University,
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan

Abstract: The software release game by Zeephongsekul and Chiera (1995) is reconsidered in the framework of two person non-zero sum game of timing. More precisely, the noisy type of software release strategies under two different criteria as well as an alternative silent type of strategy are derived in the closed forms. Our method employed in this paper has an advantage over Zeephongsekul and Chiera (1995) since it is simpler and much tractable on computation of the silent strategy. Also, the method can be extended directly to obtain the noisy type of strategies.

0. Introduction

Software reliability is one of the most important issues for realization of a highly reliable computer system. As conventional methods for estimating and predicting the reliability of a developed software product, a large number of software reliability growth models based on probability and/or statistical theory have been proposed in the literature, and at the same time, have been recognized to be useful for assessing the software reliability quantitatively (see Musa, Iannino and Okumoto (1987), Xie (1991) and Lyu (1996)). These software reliability growth models have several advantages to support a decision making in the development management of software products. During the software testing phase a software system will be executed with a sampling of some test cases in order to detect/correct software faults which cause software failures. A software failure is defined as an unacceptable output of program operation caused by a software fault remaining in the system. Ordinarily, we can assume in the software reliability growth models that the correction of faults does not introduce any new fault. Though the software test has to be executed sufficiently to deliver a reliable software system to the market or the user, it should be noted that the longer software test causes the increasing software cost. Of course, if many number of software faults are remained in the software system after being released, the software developer will suffer seriously financial damages to correct them in the operation phase. Thus, it is of great importance to determine the economic timing to release the software product. Such a problem is called *the software release problem* and has been discussed by many authors.

The software release problem was first considered by Okumoto and Goel (1980). Assuming the so-called exponential software reliability growth model based on a non-homogeneous Poisson process, they formulated the total expected software cost and derived the optimal software release time to minimize it. Koch and Kubat (1983), Yamada and Osaki (1985, 1987) and Ohtera and Yamada (1990) analyzed the similar problems under several kinds of software reliability growth models and/or cost criteria. The other approaches were developed by Forman and Singpurwalla (1977, 1979), Musa and Ackerman (1989) and Ross (1985), though they never aim to get the cost-oriented software release schedules. Latter, the software release problems from the economical point of view were generalized by Dalal and Mallows (1988, 1990, 1992a, 1992b). They reformulated the underlying problem as an optimal stopping problem. In this way, most existing software release problems have been concentrated their attention to direct minimization of the total expected software cost, but have not been considered about a competitive market situation in which one can observe in real world. Abdel-Ghaly, Chan and Littlewood (1986) proposed evaluation methods of competing software reliability predictions and applied statistical plotting methods to compare two different software reliability growth models.

Recently, Zeephongsekul and Chiera (1995) and Zeephongsekul (1996) analyzed the software release problems under competitive market circumstance in the frame work of two person non–zero sum games and provided more realistic and sophisticated software release schedules. More specifically, Zeephongsekul and Chiera (1995) dealt with the software release problem as a typical but somewhat different two-person game of timing from classical ones by Fox and Kimeldorf (1969), Teraoka (1976, 1979, 1983a, 1983b), Sakaguchi (1978) and Kurisu (1983, 1989). In this paper, we call such a game–theoretic software release problem *the software release game.* It will be important to discuss the software release game in details, since the existing software release problems treat the optimization ones for only one software developer and are self-concluded problems. In other words, while most earlier papers never take the real market fluctuation into account, the factor of marketing seems to be indispensable to the evaluation. From this reason, the seminal works by Zeephongsekul and Chiera (1995) and Zeephongsekul (1996) should be encouraged and will give an impact to the software development management. Unfortunately, notice that their approach is quite complex and involves some computational complexity, as described latter. The main purpose of this paper is to reconsider the software release game by Zeephongsekul and Chiera (1995). More precisely, the noisy type of software release strategies under two different criteria as well as an alternative silent type of strategy are derived in the closed forms. Our method employed in this paper has an advantage over Zeephongsekul and Chiera (1995) since it is simpler and much tractable on computation of the silent strategy. Also, the method can be extended directly to obtain the noisy type of strategies. In the rest part of this section, we introduce the most basic software release problem by Okumoto and Goel (1980) and prepare to derive the game-theoretic strategies in sequential sections.

Suppose that there exist only two software developers in the market and that they share it. For convenience, they are labeled as Player 1 and Player 2, respectively. Let $\{N_i(t), 0 \leq t \leq T_{LC}, i = 1, 2\}$ be the stochastic point processes representing the number of faults experienced in a software program up to time t in the testing phase, where T_{LC} (> 0) denotes the software life cycle and is a pre-determined constant. Without any loss of generality, we assume that $N_i(t)$ is increasing in t with $N_i(0) = 0$, $i = 1, 2$. As Chen and Singpurwalla (1997) and Al-Mutairi, Chen and Singpurwalla (1998) pointed out, it should be noted that almost software reliability growth models proposed in the literature are belonging to a sub-family of point processes. Define the following notation;

p_i: the unit price of the software product developed by Player i ($= 1, 2$)

$C_i(t)$: the total expected software cost for Player i when the software is released at time t

c_{1i}: the testing cost per unit time incurred in testing phase to Player i,

c_{2i}: the cost to remove a fault in testing phase to Player i,

c_{3i}: the cost to remove a fault in operation phase to Player i, where $c_{3i} > c_{2i}$.

If Player 1 wishes to maximize his or her own total expected profit $g_1(t) = p_1 A_1(t) - C_1(t)$, where $C_1(t) = c_{11}t + c_{21}\mathrm{E}[N_1(t)] + c_{31}\{\mathrm{E}[N_1(T_{LC})] - \mathrm{E}[N_1(t)]\}$ and $g_1(t)$ is differentiable with respect to t, then the problem is reduced to a simple algebraic one to obtain τ_1, where $\tau_i \in [0, T_{LC}] \equiv \{t \geq 0 \mid \sup g_i(t), i = 1, 2\}$ and the success in selling the product released at arbitrary time t (≥ 0) is achieved with probability $A_i(t)$. If each player does not take account of his or her opponent as the classical software release problem, it means that $A_i(t) = 0$ and $\tau_i = \gamma_i \equiv \{t \geq 0 \mid \inf C_i(t), i = 1, 2\}$. Hence if the function $\mathrm{E}[N_i(t)]$ is strictly concave in t, then $d^2 g_i(t)/dt^2 < 0$ under $c_{3i} > c_{2i}$ and there exists unique optimal software release time γ_i under the conditions that $c_{1i}/(c_{3i} - c_{2i}) < \lim_{t \to 0}(d\mathrm{E}[N_i(t)]/dt)$ and $c_{1i}/(c_{3i} - c_{2i}) > \lim_{t \to T_{LC}}(d\mathrm{E}[N_i(t)]/dt)$. To avoid trivial cases (*i.e.* $\gamma_i = 0$ and $\gamma_i = T_{LC}$), we assume that there exists unique $\gamma_i \in (0, T_{LC})$, where $\lim_{t \to \gamma_i}(d\mathrm{E}[N_i(t)]/dt) = c_{1i}/(c_{3i} - c_{2i})$, $i = 1, 2$. Of course, note that the strategy above does not take account of the effect of opponent's action in competitive environment.

The paper is organized as follows. In Section 1, we introduce the software release game by Zeephongsekul and Chiera (1995), which is a silent type game of timing viz. infinite games on the square, and point out the problems. In Section 2, a different strategy for the software release game is derived and the problems in Zeephongsekul and Chiera (1995) are overcame. Then, the concept of Nash equilibrium (1951) is useful to calculate a mixed equilibrium strategy for competitive two rival software developers. In Section 3, two noisy type of games are considered and the optimal strategies are characterized.

1. The software release game

We formulate a software release game in a fashion similar to Zeephongsekul and Chiera (1995). The software products produced by two rival players are assumed to perform virtually the same set of tasks and have the same (and constant) software life cycle T_{LC} (> 0). If the market is not competitive, they will release their softwares at the best timing which minimize the expected total software costs incurred and estimated during both the testing and operation phases, respectively. However, since the market is competitive in practice, both players will be influenced by his or her opponent and the release plans based on only their own criteria are not always adequate from the game-theoretic points of view. We suppose that Player 1 (2) has to decide to release the software product at time $t = x(y)$ and that success in selling the product released at arbitrary time t (≥ 0) is achieved with probability $A_1(t)(A_2(t))$, where the functions $A_i(t)$ are known each other from past experiences, unimodal in t with $A_i(0) = A_i(T_{LC}) = 0$, $i = 1, 2$, differentiable with respect to t, and attain their global maxima at η_i, that is, $\eta_i \in [0, T_{LC}] \equiv \{t \geq 0 \mid \sup A_i(t), i = 1, 2\}$. The unimodality of $A_i(t)$ implies that the software released at both earlier and latter stages has a smaller success probability in the market since it involves a larger number of faults at earlier phase and the success probability should become smaller in latter phase. This point is quite different from existing games of timing (Fox and Kimeldorf (1969), Teraoka (1976, 1979, 1983a, 1983b), Sakaguchi (1978) and Kurisu (1983, 1989)). For these technical assumptions, see Zeephongsekul and Chiera (1995).

Let $(X, Y) = [0, T_{LC}] \times [0, T_{LC}]$ denotes the sets of pure strategies for Player 1 and 2, respectively, and $(x, y) \in (X, Y)$ the pure strategies representing the release times for respective software products. Since the solution for two person non-zero sum game of timing is a randomized one, we adopt the mixed strategies $F_1 = F_1(x) = \Pr\{X \leq x\} \in [0, 1]$ and $F_2 = F_2(y) = \Pr\{Y \leq y\} \in [0, 1]$ (see Nash (1951) and Fox and Kimeldorf (1969)). This means that Player i release the software products at random times characterized by probability distribution functions F_i, $i = 1, 2$. More specifically, if Player i take the mixed strategies F_i, then their total expected software rewards can be represented as $M_i(F_1, F_2)$, $i = 1, 2$. Zeephongsekul and Chiera (1995) formulated the total expected software rewards for respective players as follows.

$$M_1(x, y) = \begin{cases} p_1 A_1(x) - C_1(x) & (0 \leq x < y \leq T_{LC}) \\ p_1 \overline{A}_2(y) A_1(x) - C_1(x) & (0 \leq y \leq x \leq T_{LC}) \end{cases} \tag{1}$$

and

$$M_2(x, y) = \begin{cases} p_2 A_2(y) - C_2(y) & (0 \leq y < x \leq T_{LC}) \\ p_2 \overline{A}_1(x) A_2(y) - C_2(y) & (0 \leq x \leq y \leq T_{LC}), \end{cases} \tag{2}$$

where, in general, $\overline{A}(\cdot) = 1 - A(\cdot)$.

Remark 1.1: It should be noted that the two-person non-zero sum game formulated above is a silent game, though Zeephongsekul and Chiera (1995) say that it is a noisy game. That is, each player can not know the opponent's action.

Define

$$M_i(F_1, F_2) = \int_X \int_Y M_i(x,y) dF_1 dF_2, \quad i = 1, 2, \tag{3}$$

where

$$M_1(x, F_2) = \int_Y M_1(x,y) dF_2 \tag{4}$$

and

$$M_2(F_1, y) = \int_X M_2(x,y) dF_1. \tag{5}$$

Then the problem is to derive *the mixed equilibrium strategies (or Nash strategies)* (F_1^*, F_2^*) or their densities $(\partial F_1^*(x)/\partial x, \partial F_2^*(y)/\partial y) = (f_1^*, f_2^*)$, if exists, satisfying

$$M_1(F_1^*, F_2^*) \geq M_1(F_1, F_2^*) \tag{6}$$

and

$$M_2(F_1^*, F_2^*) \geq M_2(F_1^*, F_2), \tag{7}$$

where $M_i(F_1^*, F_2^*)$, $i = 1, 2$, is called *the game value* or simply *value function.*

The following lemma which was obtained by Zeephongsekul and Chiera (1995) is needed to complete the discussion throughout this paper.

Lemma 1.2: $\eta_i \wedge \gamma_i \leq \tau_i \leq \eta_i \vee \gamma_i$.

Zeephongsekul and Chiera (1995) derived the mixed equilibrium strategies satisfying Eqs. (6) and (7). We give the following result without proof.

Proposition 1.3: The mixed equilibrium strategies by Zeephongsekul and Chiera (1995)

$$F_i^*(t) = \begin{cases} 0, & 0 \leq t < \rho_{3-i} \\ \int_{\rho_{3-i}}^t f_i(y)dy, & \rho_{3-i} \leq t \leq \sigma_{3-i} \\ 1, & \sigma_{3-i} < t \leq T_{LC} \end{cases} \tag{8}$$

satisfy the inequalities in Eqs (6) and (7), where $\rho_i, i = 1, 2$, *is the unique root of*

$$\int_{\rho_i}^{\sigma_i} f_{3-i}(y)dy = 1 \tag{9}$$

and where

$$f_i(t) = \frac{\{C_{3-i}(t) + g_{3-i}(\rho_{3-i})\}(\partial A_{3-i}(t)/\partial t) - (\partial C_{3-i}(t)/\partial t)A_{3-i}(t)}{p_{3-i}A_i(t)\{A_{3-i}(t)\}^2}, \tag{10}$$

and $\sigma_i = \sup\{t : f_{3-i}(t) > 0\}$.

Remark 1.4: Zeephongsekul and Chiera (1995) proposed the above mixed equilibrium strategies in the silent game setting. However, it should be noted that their strategies have some computational complexity. For instance, in order to derive $f_{3-i}(t)$, one has to set an initial value for ρ_{3-i} and calculate $t = \sigma_i$ so as to satisfy $\sup\{t : f_{3-i}(t) > 0\}$. Since the value of σ_i does not always satisfy Eq.(9), one must calculate the next candidate of ρ_i for the fixed σ_i. Then the same procedure is repeated until both conditions $\int_{\rho_i}^{\sigma_i} f_{3-i}(y)dy \approx 1$ and $f_{3-i}(\sigma_i) \approx 0$ are satisfied. However, such a iteration method does not always guarantee the uniqueness of the solution, and in fact is not feasible for some parameter setting.

2. An alternative silent strategy

In this section, we consider a different silent type of game from Zeephongsekul and Chiera (1995) and give an alternative mixed equilibrium strategy. Zeephongsekul and Chiera (1995) construct $F_i^*(t)$ with ρ_i satisfying the normal condition $\int_{\rho_i}^{\sigma_i} f_{3-i}(y)dy = 1$ for a fixed σ_1, but we determine the parameter a so as to satisfy $\int_a^{\eta} f_i(y)dy = 1$ with $\eta = \min(\eta_1, \eta_2)$. In that sense, our stopping rule is expected to be quite different from Eq.(8) and much simple for calculation.

The necessary condition to exist the optimal software release strategy is to exist the solution of the first-order condition of optimality;

$$\frac{\partial}{\partial x} M_1(x, F_2) = 0 \tag{11}$$

for Player 1. On the other hand, the sufficient condition is to show the existence of the game value $M_i(F_1^*, F_2^*), i = 1, 2$, satisfying Eqs. (6) and (7). In the first part of this section, let us consider the necessary condition. Solving the differential equation above, the first-order condition of optimality for mixed equilibrium strategies $f_i(x)$ can be derived as follows.

Lemma 2.1: The solutions of the differential equations in Eq.(11) are

$$f_i(t) = \begin{cases} 0, & (0 \leq t < a), \\ \dfrac{(\partial A_{3-i}(t)/\partial t)\{C_{3-i}(t) + g_{3-i}(a)\} - (\partial C_{3-i}(t)/\partial t)A_{3-i}(a)}{p_{3-i}\{A_{3-i}(t)\}^2 A_i(t)}, & (a \leq t \leq \eta), \end{cases} \tag{12}$$

where a_i, $i = 1, 2$, satisfies $\int_{a_i}^{\eta} f_i(x)dx = 1$, $a = \max(a_1, a_2)$ and $\eta = \min(\eta_1, \eta_2)$.

Lemma 2.2: $0 < a < \eta \leq \tau_i$, $i = 1, 2$, if $f_i(x) \geq 0$ for $a \leq x \leq \eta$.

Lemma 2.3: There exists unique a (> 0) such that

$$\int_a^{\eta} f_i(t)dt \geq 1, \quad i = 1, 2. \tag{13}$$

Now, we are in the position to derive the mixed equilibrium strategies. Since the following theorem is due to Teraoka (1979, 1983a,b), the proof is omitted for brevity.

Theorem 2.4: The mixed equilibrium strategies for player i $(= 1, 2)$ are

$$F_1^*(x) = \int_a^{\eta} f_1(x)dx + \alpha I(z) \tag{14}$$

and

$$F_2^*(y) = \int_a^{\eta} f_2(y)dy + \beta I(z), \tag{15}$$

where $I(z) = 1$ at $z = \eta$, otherwise, $I(z) = 0$, and the parameters α and β are

$$\alpha \left\{ \begin{array}{c} = \\ = \\ > \end{array} \right\} 0, \quad \beta \left\{ \begin{array}{c} > \\ = \\ = \end{array} \right\} 0, \quad a = \left\{ \begin{array}{c} a_1 > a_2 \\ a_1 = a_2 \\ a_1 < a_2 \end{array} \right\}, \tag{16}$$

$$\alpha = 1 - \int_a^\eta f_1(x)dx, \quad \beta = 1 - \int_a^\eta f_2(y)dy. \tag{17}$$

Theorem 2.5: The values of software release game are

$$M_i(F_1^*, F_2^*) = g_i(a), \quad i = 1, 2 \tag{18}$$

and the optimal mixed equilibrium strategies are given by $F_i^*(t)$ in Eqs.(14) and (15).

Remark 2.6: Note that Theorems 2.4 and 2.5 correspond to necessary and sufficient conditions, respectively, for the mixed equilibrium strategies. From Theorem 2.4, in particular, our strategies involve only one unknown parameter a and overcome the incompleteness of computational procedure in Zeephongsekul and Chiera (1995). Further, this motivates that the noisy type of game should be considered in the following section.

3. Noisy games

(3.1) Noisy Game A

The first part of this section concerns the noisy version of the previous section. That is, two players can observe opponent's action each other and can release the software products at the best timing if the opponent has already released and failed in the market, where "the best timing" means the time instant when their total expected software rewards are maximized. Then, we have

$$M_1(x,y) = \begin{cases} p_1 A_1(x) - C_1(x) & (0 \leq x < y \leq T_{LC}) \\ \bar{A}_2(y)\max\{p_1 A_1(x) - C_1(x)\} & (0 \leq y \leq x \leq T_{LC}) \end{cases} \tag{19}$$

and

$$M_2(x,y) = \begin{cases} p_2 A_2(y) - C_2(y) & (0 \leq y < x \leq T_{LC}) \\ \bar{A}_1(x)\max\{p_2 A_2(y) - C_2(y)\} & (0 \leq x \leq y \leq T_{LC}). \end{cases} \tag{20}$$

To derive the mixed equilibrium strategies satisfying Eqs. (6) and (7), we obtain the first-order conditions of optimality which satisfy $\partial M_1(x, F_2)/\partial x = 0$ and $\partial M_2(F_1, y)/\partial y = 0$ as follows.

Lemma 3.1: The solutions of the differential equations $\partial M_1(x, F_2)/\partial x = 0$ and $\partial M_2(F_1, y)/\partial y = 0$ are

$$f_i(t) = \begin{cases} 0 & (0 \leq t < a) \\ k_{3-i}^A \exp\left\{\int D_{3-i}^A(t)dt\right\} & (a \leq t \leq \tau), \end{cases} \tag{21}$$

where, for $i = 1, 2$,

$$D_i^A(t) = \frac{1 - (\partial B_i^A(t)/\partial t)}{B_i^A(t)}, \tag{22}$$

$$B_i^A(t) = \frac{g_i(\tau) - A_{3-i}(t)g_i(\tau) - g_i(t)}{(\partial g_i(t)/\partial t)} \tag{23}$$

and

$$k_i^A = \frac{\lim_{t\to a}(\partial g_i(t)/\partial t)}{g_i(\tau) - A_{3-i}(a)g_i(\tau) - g_i(a)} \exp\{-\int D_i^A(a)da\}. \tag{24}$$

and where a_i, $i = 1, 2$, *satisfies* $\int_{a_i}^{\tau} f_i(t)dt = 1$ *with* $a = \max(a_1, a_2)$ *and* $\tau = \min(\tau_1, \tau_2)$.

Lemma 3.2: $0 < a < \tau$ *and* $g_i(a) \geq \overline{A}_{3-i}(a)g_i(\tau), i = 1, 2$.

Lemma 3.3: Define $Q_i^A(t) \equiv \exp\{\int D_i^A(t)dt\}, i = 1, 2$. *If* $\{1 + \overline{A}_i(a)\}g_i(\tau)/2 \ \geq \ (<) \ g_i(a)$ *and the function* $Q_i^A(t)$ *increases (strictly decreases) in* t, *then*

$$\int_{a_{3-i}}^{\tau} f_{3-i}(t)dt \ \geq \ (<) \ 1. \tag{25}$$

Theorem 3.4: (i) If $\{1 + \overline{A}_i(a)\}g_i(\tau)/2 \geq g_i(a)$ *and* $Q_i^A(t)$ *increases in* t, *then*

$$F_1^*(x) = \begin{cases} 0 & (0 \leq x < a) \\ \int_a^{\tau} f_1(x)dx + \alpha I(z) & (a \leq x \leq \tau) \end{cases} \tag{26}$$

and

$$F_2^*(y) = \begin{cases} 0 & (0 \leq y < a) \\ \int_a^{\tau} f_2(y)dy + \beta I(z) & (a \leq y \leq \tau) \end{cases} \tag{27}$$

where $I(z) = 1$ *at* $z = \tau$, *otherwise,* $I(z) = 0$, *the parameters* α *and* β *satisfy Eq.(16) and*

$$\alpha = 1 - \int_a^{\tau} f_1(x)dx, \quad \beta = 1 - \int_a^{\tau} f_2(y)dy. \tag{28}$$

(ii) If $\{1 + \overline{A}_i(a)\}g_i(\tau)/2 < g_i(a)$ *and* $Q_i^A(t)$ *strictly decreases in* t, *then the mixed equilibrium strategies are given by Eqs.(26) and (27), where* $\alpha > 0$ *and* $\beta > 0$.

Theorem 3.5: Suppose that either condition (i) or (ii) of Theorem 3.4 is satisfied. The values of software release game are

$$M_i(F_1^*, F_2^*) = g_i(a), \quad i = 1, 2. \tag{29}$$

(3.2) Noisy Game B

Next, we consider the different noisy type of software release game. Suppose that two players can observe opponent's action each other and wish to release the software products at the timing when their success probabilities in the market are maximized. Of course, this problem is quite different from the noisy game A. Consider the following total expected software reward:

$$M_1(x, y) = \begin{cases} p_1 A_1(x) - C_1(x) & (0 \leq x < y \leq T_{LC}) \\ \overline{A}_2(y)\{p_1 \max A_1(x)\} - C_1(x) & (0 \leq y \leq x \leq T_{LC}) \end{cases} \tag{30}$$

and

$$M_2(x,y) = \begin{cases} p_2 A_2(y) - C_2(y) & (0 \le y < x \le T_{LC}) \\ \overline{A}_1(x)\{p_2 \max A_2(y)\} - C_2(y) & (0 \le x \le y \le T_{LC}). \end{cases} \tag{31}$$

Lemma 3.6: The solutions of the differential equations $\partial M_1(x, F_2)/\partial x = 0$ and $\partial M_2(F_1, y)/\partial y = 0$ *for Eqs. (30) and (31) are*

$$f_i(t) = \begin{cases} 0 & (0 \le t < a) \\ k_{3-i}^B \exp\left\{\int D_{3-i}^B(t)dt\right\} & (a \le t \le \eta) \end{cases} \tag{32}$$

where

$$D_i^B(t) = \frac{1 - (\partial B_i^B(t)/\partial t)}{B_i^B(t)}, \tag{33}$$

$$B_i^B(t) = \frac{g_i(\eta) - A_{3-i}(t)g_i(\eta) - g_i(x)}{(\partial g_i(t)/\partial t)}, \tag{34}$$

$$k_i^B = \frac{\lim_{t\to a}(\partial g_i(t)/\partial t)}{g_i(\eta) - A_{3-i}(a)g_i(\eta) - g_i(a)} \exp\{-\int D_i^B(a)da\} \tag{35}$$

and, where a_1 *and* a_2 *are the unique root of*

$$\int_{a_i}^{\eta} f_i(t)dt = 1, \quad i = 1, 2, \tag{36}$$

$a = \max(a_1, a_2)$ *and* $\eta = \min(\eta_1, \eta_2)$.

Lemma 3.7: $0 < a < \eta$ *and* $g_i(a) > \overline{A}_{3-i}(a)g_i(\eta), i = 1, 2.$

Lemma 3.8: Define $Q_i^B(t) \equiv \exp\left\{\int D_i^B(t)dt\right\}, i = 1, 2$. *If* $\{1 + \overline{A}_i(a)\}g_i(\eta)/2 \ge (<) g_i(a)$ *and the function* $Q_i(t)$ *increases (strictly decreases) in* t, *then*

$$\int_{a_{3-i}}^{\eta} f_{3-i}(x)dx \ \ge \ (<) \ 1. \tag{37}$$

Theorem 3.9: (i) If $\{1 + \overline{A}_i(a)\}g_i(\eta)/2 \ge g_i(a)$ *and* $Q_i^B(t)$ *increases in* t, *then*

$$F_1^*(x) = \begin{cases} 0 & (0 \le x < a) \\ \int_a^{\eta} f_1(x)dx + \alpha I(z) & (a \le x \le \eta), \end{cases} \tag{38}$$

and

$$F_2^*(y) = \begin{cases} 0 & (0 \le y < a) \\ \int_a^{\eta} f_2(y)dy + \beta I(z) & (a \le y \le \eta), \end{cases} \tag{39}$$

where $I(z) = 1$ *at* $z = \eta$, *otherwise,* $I(z) = 0$, *the parameters* α *and* β *satisfy Eq. (16) and*

$$\alpha = 1 - \int_a^\eta f_1(x)dx, \quad \beta = 1 - \int_a^\eta f_2(y)dy. \tag{40}$$

(ii) If $\{1 + \overline{A}_i(a)\}g_i(\eta)/2 < g_i(a)$ *and* $Q_i^B(t)$ *strictly decreases in* t, *then the mixed equilibrium strategies are given by Eqs.(38) and (39), where* $\alpha > 0$ *and* $\beta > 0$.

Theorem 3.10: Suppose that either condition (i) or (ii) of Theorem 3.9 is satisfied. The values of software release game are

$$M_i(F_1^*, F_2^*) = g_i(a), \quad i = 1, 2. \tag{41}$$

References

Abdel-Ghaly, A. A., Chan, P. Y. and Littlewood, B. (1986) Evaluation of competing software reliability predictions. *IEEE Transactions on Software Engineering* **SE-12** 950-967.

Al-Mutairi, D., Chen, Y. and Singpurwalla, N. D. (1998) An adaptive concatenated failure rate model for software reliability. *Journal of American Statistical Association* **93**, 1150-1163.

Chen, Y. and Singpurwalla, N. D. (1997) Unification of software reliability models by self-exciting point processes. *Advances in Applied Probability* **29**, 337-352.

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

Dalal, S. R. and Mallows, C. L. (1990) Some graphical aids for deciding when to stop testing software. *IEEE Journal of Selected Areas in Communications* **8**, 169-175.

Dalal, S. R. and Mallows, C. L. (1992a) When to stop testing software – some exact and asymptotic results. *Bayesian Analysis in Statistics and Econometrics, Lecture Notes in Statistics* **75**, Springer-Verlag, New York, 267-276.

Dalal, S. R. and Mallows, C. L. (1992b) Buying with exact confidence. *Annals of Applied Probability* **2**, 752-765.

Forman, E. H. and Singpurwalla, N. D. (1977) An empirical rule for debugging and testing computer software. *Journal of American Statistical Association* **72**, 750-757.

Forman, E. H. and Singpurwalla, N. D. (1979) Optimal time intervals for testing hypotheses on computer software errors. *IEEE Transactions on Reliability* **R-28**, 250-253.

Fox, M. and Kimeldorf, G. (1969) Noisy duels. *SIAM Journal of Applied Mathematics* **17**, 353-361.

Koch, H. S. and Kubat, P. (1983) Optimal release time for computer software. *IEEE Transactions on Software Engineering* **SE-9**, 323-327.

Kurisu, T. (1983) On the noisy-silent versus silent-noisy duel with equal accuracy functions. *Journal of Optimization Theory and Applications* **40**, 175-199.

Kurisu, T. (1989) On a duel with time lag and equal accuracy functions. *Journal of Optimization Theory and Applications* **60**, 43-56.

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

Musa, J. D. and Ackerman, A. F. (1989) Quantifying software validation: when to stop testing. *IEEE Software* **6**, 19-27.

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

Nash, J. (1951) Non-cooperative games. *Annals of Mathematics* **54**, 286-295.

Ohtera, H. and Yamada, S. (1990) Optimum software release time considering an error detection phenomenon during operation. *IEEE Transactions on Reliability* **R-39**, 596-599.

Okumoto, K. and Goel, A. L. (1980) Optimum release time for software systems based on reliability and cost criteria. *Journal of Systems and Software* **1**, 315-318.

Ross, S. M. (1985) Software reliability: the stopping rule problem. *IEEE Transactions on Software Engineering* **SE-11**, 1472-1476.

Sakaguchi, M. (1978) Marksmanship contests – non-zero sum game of timing. *Mathematica Japonica* **22**, 585-596.

Teraoka, Y. (1976) Noisy duel with uncertain existence of the shot. *International Journal of Game Theory* **5**, 239-249.

Teraoka, Y. (1979) A two-person game of timing with random arrival time of the object. *Mathematica Japonica* **24**, 427-438.

Teraoka, Y. (1983a) A two-person infinite game suggested from a quiz of guessing a number. *Bulletin of Informatics and Cybernetics* **20**, 23-32.

Teraoka, Y. (1983b) A two-person game of timing with random termination. *Journal of Optimization Theory and Applications* **40**, 379-396.

Xie, M. (1991) *Software Reliability Modelling,* World Scientific, Singapore.

Yamada, S. and Osaki, S. (1985) Cost-reliability optimal release policies for software systems. *IEEE Transactions on Reliability* **R-34**, 422-424.

Yamada, S. and Osaki, S. (1987) Optimal software release policies with simultaneous cost and reliability requirements. *European Journal of Operational Research* **31**, 46-51.

Zeephongsekul, P. and Chiera, C. (1995) Optimal software release policy based on a two-person game of timing. *Journal of Applied Probability* **32**, 470-481.

Zeephongsekul, P. (1996) Stackelberg strategy solution for optimal software release policies. *Journal of Optimization Theory and Applications* **91**, 215-233.