

2 変量ソフトウェア信頼度成長モデルの開発（継続）

代表研究者 南野友香 鳥取大学大学院工学研究科社会経営工学講座 助教

1 はじめに

ソフトウェア開発におけるテスト工程は、ソフトウェアの最終的な品質/信頼性を計測・評価するための重要な工程であり、CPU 時間、システムの実行時間、工数、テスト網羅度、実行されたテストケース数といった、いわゆるテスト労力が莫大に投入されている。また、ソフトウェア品質/信頼性の定量的評価手法の基盤技術としては、ソフトウェア信頼度成長モデル (SRGM: software reliability growth model) [2, 10] がよく知られている。これは、ソフトウェア実行過程に対するフォールト発見事象や、ソフトウェア故障発生現象をソフトウェアの信頼度成長として扱う確率モデルである。しかしながら、実際のテスト工程で莫大なテスト労力が投入されているにも関わらず、従来の SRGM はテスト時間のみに依存すると仮定されている。したがって、テスト労力の投入過程を考慮できていないため、従来の SRGM は、より正確にフォールト発見過程を捉えることができていないと考えられる。

このような背景から、本研究では、テスト時間およびテスト労力を考慮した 2 変量 SRGM を構築する。まず、信頼度成長要因としてのテスト時間をテスト時間要因とテスト労力要因に大別し、テスト時間関数を仮定する。テスト時間関数には、経済学における生産関数としてよく用いられるコブ・ダグラス型関数 [6, 7, 9] および CES (constant elasticity of substitution) 型関数 [11] を用いる。コブ・ダグラス型関数は、非常にシンプルで扱いやすい関数形であり、CES 型関数は、コブ・ダグラス型関数を含む種々の生産関数を一般化した関数である。これらの関数を用いて、テスト時間要因およびテスト労力要因から成るテスト時間関数を仮定し、従来の指数形、遅延 S 字形、および習熟 S 字形 SRGM に導入することで、2 変量 SRGM に拡張する。モデル内のパラメータは、実測データを用いて最尤法により推定する。このとき、統計解析フリーソフトの R を用いる。その後、提案モデルの有効性を検証するため、MSE (平均偏差平方和, mean squared errors) による従来モデルとの適合性比較を行う。

さらに、ソフトウェア開発管理面への応用問題として、提案モデルに基づくソフトウェア最適リリース問題 [8, 10] について議論する。ソフトウェア最適リリース問題とは、QCD (Q: quality, C: cost, D: delivery) を考慮しながら、ソフトウェアをテスト工程から運用段階へ移行させる最適な時刻を求める最適化問題のことをいう。本研究では、コブ・ダグラス型テスト時間関数に基づく 2 変量指数形 SRGM を適用し、総期待ソフトウェアコストを最小にする最適リリース時刻および最適テスト労力量を解析的に推定する。最後に、実測データを用いて数値例を示す。

2 テスト時間関数を考慮した 2 変量ソフトウェア信頼度成長モデル

2-1 従来モデル

検出可能なフォールト数が有限の場合、多くの非同次ポアソン過程 (NHPP: nonhomogeneous Poisson process) モデルでは、単位時間あたりに発見されるフォールト数はその時点でソフトウェア内に残存するフォールト数に比例するものと仮定される。この仮定に基づき構築された代表的な NHPP モデルは、それぞれ次式のように表される [10]。

指数形 SRGM (EXP model) :

$$H(t) \equiv m(t) = a(1 - \exp[-bt]). \quad (1)$$

遅延 S 字形 SRGM (DSS model) :

$$H(t) \equiv M(t) = a(1 - (1 + bt)\exp[-bt]). \quad (2)$$

習熟 S 字形 SRGM (ISS model) :

$$H(t) \equiv I(t) = \frac{a(1 - \exp[-bt])}{(1 + c \cdot \exp[-bt])}. \quad (3)$$

ここで, $a(> 0)$ は初期潜在フォールト数, $b(> 0)$ は 1 個当たりのフォールト発見率, および $H(t)$ は NHPP の平均値関数であり, 時間区間 $(0, t]$ において発見される総期待フォールト数を表す. また, $c = (1 - l)/l$ であり, $l (0 < l \leq 1)$ はフォールト発見能力に関するテスト習熟係数を表す.

2-2 コブ・ダグラス型テスト時間関数に基づく 2 変量 SRGM

まず, ソフトウェア信頼度成長要因としてのテスト時間をテスト時間要因 (s) とテスト労力要因 (u) に大別する. テスト時間要因はカレンダー時間であり, テスト労力要因は, CPU 時間, 工数, テスト網羅度, および実行されたテストケース数等であるとする. したがって, 従来モデルにおけるテスト時刻 t は, コブ・ダグラス型関数により, 次式のように表される[6, 9].

$$t \equiv s^\alpha u^{1-\alpha} \quad (0 \leq \alpha \leq 1). \quad (4)$$

α はソフトウェア信頼度成長要因の影響度合いを表す分配パラメータである. 次に, 2 変量非同次ポアソン過程[6, 8]を仮定し, 式(4)のコブ・ダグラス型テスト時間関数を式(1)~(3)の NHPP モデルに適用すると, 2 変量 SRGM はそれぞれ次式のように表される[11].

2 変量指数形 SRGM (C-D EXP model) :

$$H(s, u) \equiv m_{CD}(s, u) = a(1 - \exp[-bs^\alpha u^{1-\alpha}]). \quad (5)$$

2 変量遅延 S 字形 SRGM (C-D DSS model) :

$$H(s, u) \equiv M_{CD}(s, u) = a(1 - (1 + bs^\alpha u^{1-\alpha})\exp[-bs^\alpha u^{1-\alpha}]). \quad (6)$$

2 変量習熟 S 字形 SRGM (C-D ISS model) :

$$H(s, u) \equiv I_{CD}(s, u) = \frac{a(1 - \exp[-bs^\alpha u^{1-\alpha}])}{(1 + c \cdot \exp[-bs^\alpha u^{1-\alpha}])}. \quad (7)$$

2-3 CES 型テスト時間関数に基づく 2 変量 SRGM

コブ・ダグラス型テスト時間関数を適用する場合と同様に, 従来モデルにおけるテスト時刻 t は, CES 型関数により次式のように表される.

$$t \equiv (\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}} \quad (\rho \leq 1). \quad (8)$$

ρ は代替パラメータを表す. CES 型関数は代替パラメータに応じて種々の関数形をとることができ, $\rho \rightarrow 1$ のときに 1 次の線形関数, $\rho \rightarrow 0$ のときにコブ・ダグラス型関数, $\rho \rightarrow \infty$ のときにレオンチェフ型関数となる. また, 代替の弾力性 e は, 代替パラメータを用いて, 次式のように表される.

$$e = \frac{1}{1 - \rho}. \quad (9)$$

2 変量非同次ポアソン過程を仮定し, 式(8)の CES 型テスト時間関数を式(1)~(3)の NHPP モデルに適用すると, 2 変量 SRGM はそれぞれ次式のように表される[11].

2 変数指数形 SRGM (CES-EXP model) :

$$H(s, u) \equiv m_{\text{CES}}(s, u) = a(1 - \exp[-b(\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}}]). \quad (10)$$

2 変数遅延 S 字形 SRGM (CES-DSS model) :

$$H(s, u) \equiv M_{\text{CES}}(s, u) = a(1 - (1 + b(\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}})\exp[-b(\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}}]). \quad (11)$$

2 変数習熟 S 字形 SRGM (CES-ISS model) :

$$H(s, u) \equiv I_{\text{CES}}(s, u) = \frac{a(1 - \exp[-b(\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}}])}{(1 + c \cdot \exp[-b(\alpha s^\rho + (1 - \alpha)u^\rho)^{\frac{1}{\rho}}])}. \quad (12)$$

2-4 ソフトウェア信頼性評価尺度

2 変数 SRGM に基づき, ソフトウェア信頼度および期待残存フォールト数を導出する. ソフトウェア信頼度とは, 時間区間 $(s, s + x](s \geq 0, x \geq 0)$ において, ソフトウェア故障が発生しない確率である. ただし, テスト労力量は, テスト終了時刻 s までに u まで達しているものとする. したがって, ソフトウェア信頼度は, 次式のように表される.

$$R(x|s, u) = \exp[-\{H(s + x, u) - H(s, u)\}]. \quad (13)$$

任意のテスト時刻 s およびテスト労力量 u における期待残存フォールト数は, 次式のように表される.

$$M(s, u) = H(\infty, \infty) - H(s, u) = a - H(s, u). \quad (14)$$

3 ソフトウェア最適リリース問題への応用

ソフトウェア最適リリース問題を議論するにあたり, 式(5)で表されるコブ・ダグラス型テスト時間関数に基づく 2 変数指数形 SRGM (C-D EXP model) を適用すると, 総期待ソフトウェアコストは次式で表される.

$$C(s, u) = c_1 m_{\text{CD}}(s, u) + c_2 \{a - m_{\text{CD}}(s, u)\} + c_3 s + c_4 u. \quad (15)$$

コストパラメータ $c_1 (c_1 > 0)$ はテスト段階で発見されるフォールト 1 個当りの修正コスト, $c_2 (c_2 > 0)$ は運用段階で発見されるフォールト 1 個当りの修正コスト, $c_3 (c_3 > 0)$ は単位時間当りのテストコスト, $c_4 (c_4 > 0)$ は単位テスト労力当りのテストコストを表す.

式(15)より, 次式を解くと, 総期待ソフトウェアコストを最小化する時刻およびテスト労力量を導出することができる.

$$\frac{\partial C(s, u)}{\partial s} = \frac{\partial C(s, u)}{\partial u} = 0. \quad (16)$$

したがって, ソフトウェア最適リリース時刻 s^* および最適テスト労力量 u^* は, それぞれ次式のように得られる.

$$s^* = -\frac{1}{b} \left[\frac{(1 - \alpha)c_3}{\alpha c_4} \right]^{\alpha-1} \ln \frac{\left[\frac{(1 - \alpha)c_3}{\alpha c_4} \right]^\alpha c_4}{(c_2 - c_1)(1 - \alpha)ab}. \quad (17)$$

$$u^* = -\frac{1}{b} \left[\frac{(1-\alpha)c_3}{\alpha c_4} \right]^\alpha \ln \frac{\left[\frac{(1-\alpha)c_3}{\alpha c_4} \right]^\alpha c_4}{(c_2 - c_1)(\alpha)ab}. \quad (18)$$

4 数値例

本研究では、実際のテスト工程で収集された次のデータを用いて数値例を示す[1, 3-5].

$$\mathbf{DS1} : (s_k, u_k, y_k)(k = 1, 2, \dots, 19; s_{19} = 19, u_{19} = 47.65, y_{19} = 328)$$

$$\mathbf{DS2} : (s_k, u_k, y_k)(k = 1, 2, \dots, 21; s_{21} = 21, u_{21} = 8736, y_{21} = 43)$$

$$\mathbf{DS3} : (s_k, u_k, y_k)(k = 1, 2, \dots, 20; s_{20} = 20, u_{20} = 10000, y_{20} = 100)$$

$$\mathbf{DS4} : (s_k, u_k, y_k)(k = 1, 2, \dots, 19; s_{19} = 19, u_{19} = 10272, y_{19} = 120)$$

$$\mathbf{DS5} : (s_k, u_k, y_k)(k = 1, 2, \dots, 12; s_{12} = 12, u_{12} = 5053, y_{12} = 61)$$

$$\mathbf{DS6} : (s_k, u_k, y_k)(k = 1, 2, \dots, 19; s_{19} = 19, u_{19} = 11305, y_{19} = 42)$$

ここで、 s_k はカレンダー時間 (週)、 u_k は CPU 時間/実行時間、および y_k は $[0, s_k]$ 、 $[0, u_k]$ までに発見された総フォールト数を表す。

図 1 に、DS1 において推定された、コブ・ダグラス型テスト時間関数に基づく 2 変量習熟 S 字形 SRGM (C-D ISS model) の挙動を示す。図 1 より、カレンダー時間および CPU 時間/実行時間の増加に伴い、総期待発見フォールト数が増加していることが確認できる。また、DS1 において、リリース後 1 週間目のソフトウェア信頼度は $\hat{R}(1.0|19, 47.65) = 2.9 \times 10^{-2}$ 、期待残存フォールト数は $\hat{M}(19, 47.65) \approx 26.0$ と推定された。

次に、表 1 に DS1~DS6 における従来モデルと提案モデルの適合性比較結果を示す。表中の太字は、指数形、遅延 S 字形、および習熟 S 字形の各モデルのなかで、最も適合性が高いモデルを表す。下線は、各データセットにおいて、すべてのモデルのなかで最も適合性が高いモデルを表す。表 1 より、指数形の場合は、従来モデルの EXP model よりも、提案モデルである C-D EXP model または CES-EXP model の適合性が高いことがわかる。遅延 S 字形の場合は、CES-DSS model, C-D DSS model, DSS model の順に適合性が高いという傾向が見られる。したがって、これらの提案モデルは、従来モデルよりも精度よくソフトウェア信頼度成長過程を記述していると考えられる。習熟 S 字形の場合は、ISS model と CES-ISS model の適合性が高いという結果が得られたが、MSE の値から CES-ISS model は非常に高い精度で実測データの挙動を捉えることができていることがわかる。

最後に、図 2 に推定された総期待ソフトウェアコストの挙動を示す。コストパラメータは、 $c_1 = 1$, $c_2 = 3$, $c_3 = 5$ 、および、 $c_4 = 5$ とそれぞれ仮定した。DS1 において、最適リリース時刻は $s^* = 45.97$ 、最適テスト労力量は $u^* = 18.97$ と推定された。このとき、総期待ソフトウェアコストは $C(s^*, u^*) = 1022.52$ となった。

5 おわりに

本研究では、テスト時間要因およびテスト労力要因を同時に考慮するため、コブ・ダグラス型および CES 型テスト時間関数を従来モデルに適用し、新たな 2 変量 SRGM を開発した。数値例では、実測データを用いて、提案モデルの有効性を示した。さらに、コブ・ダグラス型テスト時間関数に基づく 2 変量指数形 SRGM を適用した場合のソフトウェア最適リリース問題について議論し、コスト評価基準により最適リリース時刻および最適テスト労力量を解析的に導出した。今後の課題として、その他の提案モデルを適用した場合のソフトウェア最適リリース問題に取り組む必要がある。また、3 つ以上の信頼度成長要因を考慮する場合は、CES 型関数ではなく、複数の生産関数を組み合わせる組み合わせ型 (入れ子型) 生産関数の使用が適切であるとされているため、従来モデルに適用可能かどうかを確認する必要がある。

表 1 : MSE に基づく適合性比較結果の結果.

	EXP	C-D EXP	CES -EXP	DSS	C-D DSS	CES -DSS	ISS	C-D ISS	CES -ISS
DS1	222.00	206.24	205.98	188.93	204.79	209.20	96.65	101.86	<u>96.61</u>
DS2	7.14	7.14	10.23	3.42	3.42	3.35	<u>1.77</u>	1.96	<u>1.77</u>
DS3	20.17	16.37	20.16	28.38	28.32	27.64	<u>11.74</u>	14.42	13.80
DS4	31.20	57.93	26.52	14.04	13.86	13.24	<u>6.55</u>	31.96	9.99
DS5	27.20	6.41	23.56	10.93	6.60	5.839	2.31	2.51	<u>2.10</u>
DS6	6.01	2.79	2.64	1.095	6.28	1.09	<u>0.95</u>	1.06	<u>0.95</u>

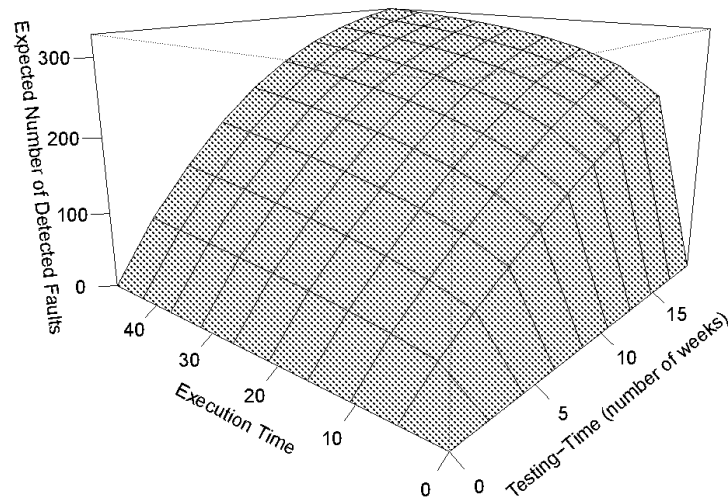


図 1: 提案モデルの挙動 (C-D ISS model, DS1, $\hat{a} = 354.12, \hat{b} = 0.17, \hat{l} = 0.23, \hat{\alpha} = 0.73$).

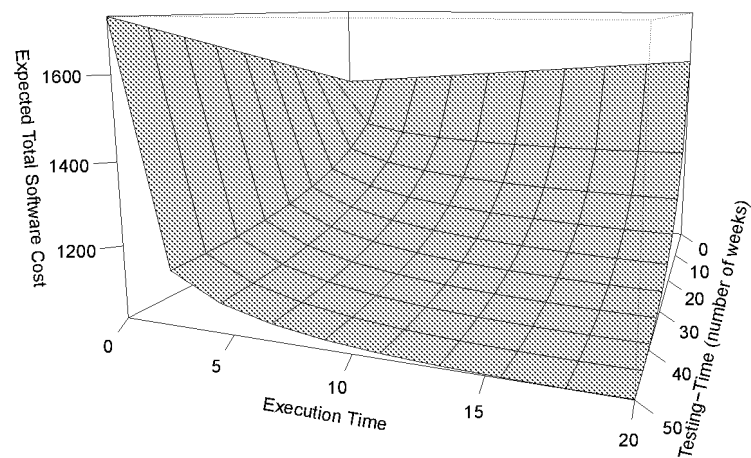


図 2: C-D ISS model を適用した場合の総期待ソフトウェアコストの挙動 (DS1, $\hat{a} = 493.24, \hat{b} = 0.044, \hat{\alpha} = 0.71, c_1 = 1, c_2 = 3, c_3 = 5, c_4 = 5$).

【参考文献】

- [1] A. Wood, "Predicting software reliability," *IEEE Computer Magazine*, Vol. 11, pp. 69-77, 1996.
- [2] H. Pham, *Software Reliability*, Springer-Verlag, Singapore, 2000.
- [3] H. Pham, "A generalized fault-detection software reliability model subject to random operating environments," *Vietnam Journal of Computer Science*, Vol. 3, Issue 3, pp. 145-150, 2016.
- [4] M. Zhu and H. Pham, "A software reliability model with time-dependent fault detection and fault removal," *Vietnam Journal of Computer Science*, Vol. 3, No. 2, pp. 71-79, 2016.
- [5] M. Ohba, "Software reliability analysis models," *IBM Journal of Research and Development*, Vol. 28, No. 4, pp. 428-443, 1984.
- [6] P.K. Kapur, A.G. Aggarwal, and G. Kaur, "Simultaneous allocation of testing-time and resources for a modular software," *International Journal of System Assurance Engineering Management*, Vol. 1, No. 4, pp. 351-361, 2010.
- [7] P.K. Kapur and H. Pham, "Two-dimensional multi-release software reliability modeling and optimal release planning," *IEEE Transactions on Reliability*, Vol. 61, No. 3, pp. 758-768, 2012.
- [8] S. Inoue, K. Fukuma, and S. Yamada, "Two-dimensional change-point modeling for software reliability assessment," *International Journal of Reliability, Quality and Safety Engineering*, Vol. 17, No. 6, pp. 531-542, 2010.
- [9] S. Inoue and S. Yamada, "Two-dimensional software reliability assessment with testing-coverage," *Proceedings of 2nd IEEE International Conference on Secure System Integration and Reliability Improvement*, pp. 150-157, 2008.
- [10] S. Yamada, *Software Reliability Modeling ---Fundamentals and Applications---*, Springer-Verlag, Tokyo/Heidelberg, 2014.
- [11] Y. Minamino, S. Inoue, and S. Yamada, "Two-dimensional software reliability growth modelling based on a CES type time function", *Proceedings of 2017 International Conference on Infocom Technologies and Unmanned Systems (ICTUS'2017)*, Dubai, U.A.E, December, 2017, pp. 120-125.

〈発表資料〉

題 名	掲載誌・学会名等	発表年月
CES型テスト時間関数に基づく2変量ソフトウェア信頼度成長モデル	京都大学数理解析研究所講究録 2078「不確実性の下での意思決定理論とその応用：計画数学の展開」	2018年7月
Bivariate software reliability models depending on a Cobb-Douglas type testing-time function and their applications	Proceedings of the 24th ISSAT International Conference on Reliability and Quality in Design	2018年8月
2変量ソフトウェア信頼度成長モデルに基づく最適リリース問題に関する研究	日本オペレーションズ・リサーチ学会中国・四国地区 SSOR アブストラクト集	2018年9月
2変量ソフトウェア信頼度成長モデリングとその応用に関する研究	第20回IEEE広島支部学生シンポジウム (HISS) 論文集	2018年11月
ソフトウェア開発管理のためのCES型テスト時間関数に基づく2変量ソフトウェア信頼性モデル	平成30年度京都大学数理解析研究所研究集会 (RIMS)「不確実性の下での数理とその周辺」	2018年11月
Two-dimensional NHPP models based on several testing-time functions and their applications	International Journal of Reliability, Quality and Safety Engineering	2019年4月