数理解析研究所講究録
第2078巻 2018年 15-21

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

南野 友香 (Yuka Minamino)† 井上 真二 (Shinji Inoue)‡ 山田 茂 (Shigeru Yamada)†

† 鳥取大学大学院・工学研究科 (Graduate School of Engineering, Tottori University)

‡ 関西大学・総合情報学部 (Faculty of Informatics, Kansai University)

1 はじめに

ソフトウェア開発工程におけるテスト工程では，最終的な品質/信頼性が定量的に計測・評価される．定量的ソフトウェア信頼性評価のための基盤的技術としては，テスト工程や運用段階におけ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過程として記述するソフトウェア信頼度成長モデル（SRGM: software reliability growth model）[4,12] が代表的である．そのなかでも，連続時間における SRGM は，主にフォールト発見数モデルとソフトウェア故障発生時間モデルに大別される．特に，フォールト発見数データは，ソフトウェア故障発生（間隔）時間データよりもデータ収集が容易であることから，フォールト発見数モデルが多く現場で利用されている [10]．しかしながら，フォールト発見数モデルをはじめとする SRGM は，テストに要した時間のみに依存すると仮定して構築されており，テスト工程におけるその他の信頼度成長要因が十分に反映されていない．したがって，これまでの SRGM では，テスト工程でどのようにテスト労力が投入されていても，テスト時間さえ長ければソフトウェア信頼度成長を観測できることを示唆している．

このような背景から，信頼度成長要因であるテスト時間要因とテスト労力要因を同時に考慮するため，コブ・ダグラス型関数を適用した 2 変量 SRGM が議論されている [3,7–11]．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，経済学においては生産関数や効用関数として用いられることが多い．しかしながら，これまでに提案されたコブ・ダグラス型関数を適用した 2 変量 SRGM において，信頼度成長要因間の代替性は考慮されていない．例えば，テスト工程において十分なテスト時間を確保できない場合，それを補うためのテスト労力が確保されていれば，ソフトウェアの信頼度成長を観測することは可能であると考えられる．一方で，テスト時間をテスト労力で代替すればするほど，代替に必要なテスト労力量が増加していくと考えられるため，代替を長期的に継続することは困難である．そのため，信頼度成長要因間で代替を行った場合，ソフトウェア信頼度成長を維持することはどの程度可能であるかを明らかにする必要がある．そこで，本研究では，経済学における代替の弾力性の概念を導入する．代替の弾力性とは，一定のソフトウェア信頼度成長を維持するために，ある信頼度成長要因を他の信頼度成長要因によって代替することの容易さを示す尺度である．コブ・ダグラス型生産関数に関しては，テスト時間要因とテスト労力要因において，代替の弾力性は常に 1 であるという性質をもつ．代替の弾力性が 1 である場合は，単位時間もしくは労力当たりのコストがどのように変化しても，総テストコストに対して信頼度成長要因の分配率は不変であることを意味する．しかしながら，代替の弾力性が 1 であるテスト環境は非常に限られているため，コブ・ダグラス型生産関数よりも柔軟に代替の弾力性の値をとることができる生産関数を用いる方が適切であると考えられる．

本研究では，コブ・ダグラス型関数よりもさらに制約が緩和され，代替の弾力性が 1 以外の一定の値をとることができる CES（constant elasticity of substitution）型関数 [1] を適用する．CES 型関数は，コブ・ダグラス型関数を含む種々の生産（効用）関数を一般化した関数であるため，他の生産関数よりも広く適用することが可能である．本研究では，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，これらの関係性を CES 型関数により表現する．これをテスト時間関数として定義し，従来の 1 変量 NHPP（nonhomogeneous Poisson process）モデル [4,12]

である指数形，遅延S字形，および習熟S字形SRGMに適用し，2変量モデルへ拡張する．その後，実測データを用いて従来モデルと提案モデルの適合性比較を行い，モデルの有効性を検証する．最後に，提案モデルにおいて推定されたパラメータに基づき，信頼度成長要因間の代替の弾力性を定量的に評価する．これにより，同じ信頼度達成目標であっても，必要に応じてテスト時間要因またはテスト労力要因に重きを置いたテスト環境を選択することができるようになると考えられる．

2 従来のソフトウェア信頼度成長モデル

検出可能なフォールト数が有限の場合，多くのNHPP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ており，その平均値関数 $H(t)$ の挙動は，

$$\frac{dH(t)}{dt} = b(t)[a - H(t)] \qquad (b(t) > 0, t \geq 0), \tag{1}$$

で表現される．このとき，a は初期潜在フォールト数，$b(t)$ は1個当りのフォールト発見率，$H(t)$ は時間区間 $(0,t]$ において発見される総期待フォールト数を表す．さらに，1個当りのフォールト発見率は次式で与えられる．

$$b(t) = \frac{\frac{d}{dt}H(t)}{[a - H(t)]}. \tag{2}$$

式(1)の微分方程式の初期条件を $H(0) = 0$ とし，$b(t) = b$ であると仮定すると，次式の指数形SRGM（EXP）が導出される．

$$H(t) \equiv m(t) = a(1 - \exp[-bt]). \tag{3}$$

式(3)において，指数形SRGMを用いて $b(t) \equiv b$，$a \equiv m(t)$ とおくと

$$\frac{dH(t)}{dt} = b[m(t) - H(t)], \tag{4}$$

のような微分方程式が得られ，これを式(3)の下で解くことにより，次のような遅延S字形SRGM（DSS）が得られる．

$$H(t) \equiv M(t) = a\big[1 - (1 + bt)\exp[-bt]\big] \qquad (a > 0, b > 0). \tag{5}$$

習熟S字形SRGMは，式(2)における $b(t)$ が，

$$b(t) = b\{l + (1 - l)\frac{H(t)}{a}\} \qquad (a > 0, b > 0, 0 < l \leq 1), \tag{6}$$

の場合を考える．ここで，l はフォールト発見能力に関するテスト習熟係数である．式(1)および式(6)を用いると，次のような習熟S字形SRGM（ISS）が得られる．

$$H(t) \equiv I(t) = \frac{a(1 - \exp[-bt])}{(1 + c \cdot \exp[-bt])} \qquad (c > 0). \tag{7}$$

ここで，$c = (1 - l)/l$ である．

3 CES型テスト時間関数に基づく2変量SRGM

本研究では，次式で表されるCES型関数[1]を適用する．

$$Y = (\alpha K^{\rho} + (1 - \alpha)L^{\rho})^{\frac{1}{\rho}} \qquad (\rho \leq 1). \tag{8}$$

ここで，経済学における企業行動を考えた場合，Y は生産性，K は資本投入量，L は労働投入量を表す．α は資本投入量と労働投入量の生産性への影響度合いを表すパラメータ（分配パラメータ），

ρ は一定の生産性を維持するために生産要素間で代替することの容易さを表現するパラメータ（代替パラメータ）である．また，CES 型関数は，1 次同次の線形関数，コブ・ダグラス型関数，およびレオンチェフ型関数を含んで一般化された関数であり，代替パラメータにおいて次の関係が成り立つ．

(1) $\rho \to 1$ のとき，1 次同次の線形関数となる．

(2) $\rho \to 0$ のとき，コブ・ダグラス型関数となる．

(3) $\rho \to \infty$ のとき，レオンチェフ型関数となる．

さらに，代替の弾力性は代替パラメータを用いて次式のように表される．

$$e = \frac{1}{1-\rho}. \tag{9}$$

本研究では，ソフトウェア信頼度成長要因としてのテスト時間が，テスト時間要因（s）とテスト労力要因（u）から成るものと定義する．具体的には，テスト時間要因はカレンダ時間（週），テスト労力要因は CPU 時間，工数，テスト網羅度，実行されたテストケース数等であるとする．以上の仮定より，従来の SRGM におけるテスト時刻 t を CES 型関数を用いて表現すると，テスト時間関数として次式のように表される．

$$t \equiv (\alpha s^{\rho} + (1-\alpha)u^{\rho})^{\frac{1}{\rho}} \qquad (\rho \leq 1). \tag{10}$$

ここで，α は信頼度成長要因としての影響度合いを表すパラメータ，ρ は一定のソフトウェア信頼度成長を維持するために信頼度成長要因間で代替を行うことの容易さを表現するパラメータであると定義する．次に，$\{N(s,u),(s \geq 0, u \geq 0)\}$ が，テスト時刻 s およびテスト労力量 u までに発見された総フォールト数を表す 2 変量確率過程であるとする．このとき，テスト時刻 s およびテスト労力量 u までに m 個のフォールトが発見される確率は，以下のように定式化される．

$$\Pr\{N(s,u)=m\} = \frac{\{H(s,u)\}^m}{m!}\exp[-H(s,u)] \qquad (m=0,1,2,\cdots). \tag{11}$$

ここで，$H(s,u)$ は時間区間 $(0,s]$ およびテスト労力量 u において発見される総期待フォールト数を表す平均値関数である．

式 (10) のテスト時間関数を式 (3)，式 (5)，および式 (7) に適用し，式 (11) の 2 変量平均値関数 $H(s,u)$ と考え，指数形（CES-EXP），遅延 S 字形（CES-DSS），および習熟 S 字形（CES-ISS）2 変量 SRGM はそれぞれ以下のように得られる．

$$H(s,u) \equiv m_{\mathrm{E}}(s,u) = a(1-\exp[-b(\alpha s^{\rho}+(1-\alpha)u^{\rho})^{\frac{1}{\rho}}]). \tag{12}$$

$$H(s,u) \equiv m_{\mathrm{D}}(s,u) = a\big(1-(1+b[\alpha s^{\rho}+(1-\alpha)u^{\rho}]^{\frac{1}{\rho}})\exp[-b[\alpha s^{\rho}+(1-\alpha)u^{\rho}]^{\frac{1}{\rho}}]\big). \tag{13}$$

$$H(s,u) \equiv m_{\mathrm{I}}(s,u) = \frac{a(1-\exp[-b(\alpha s^{\rho}+(1-\alpha)u^{\rho})^{\frac{1}{\rho}}])}{(1+c\cdot\exp[-b(\alpha s^{\rho}+(1-\alpha)u^{\rho})^{\frac{1}{\rho}}])}. \tag{14}$$

4 信頼性評価尺度

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はソフトウェア信頼度（software reliability）と呼ばれ，式 (15) のように表される．ただし，テスト労力量は，テスト終了時刻 s までに u に達しているものとする．

$$R(x|s,u) = \exp[-\{H(s+x,u)-H(s,u)\}]. \tag{15}$$

また，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．

$$M(s,u) = H(\infty,\infty) - H(s,u) = a - H(s,u). \tag{16}$$

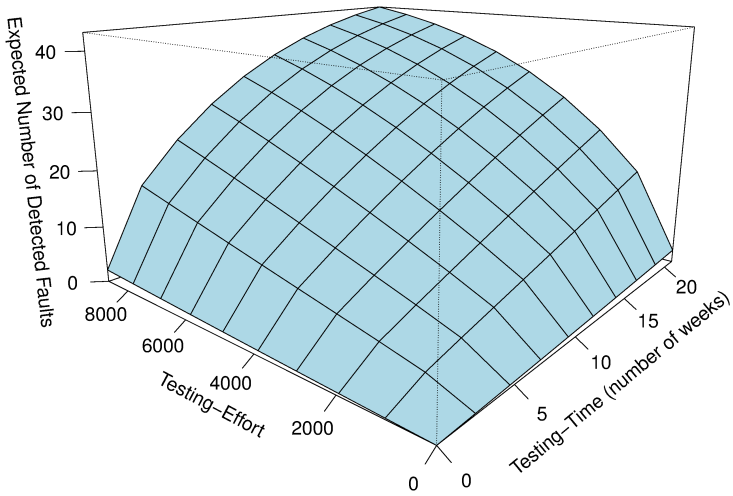

図 1： DS2 における提案モデルの挙動.（CES-ISS, $\hat{a} = 47.99$, $\hat{b} = 0.036$, $\hat{l} = 0.10$, $\hat{\alpha} = 0.85$, $\hat{\rho} = 0.283$）

5 数値例

本研究では，以下の実測データ $(s_k, u_k, y_k)(k = 0, 1, 2, \cdots, K)$ [2,5,6] を用いて数値例を示す.

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

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

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

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

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

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

ここで，s_k はカレンダ時間（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．DS3〜DS6 は同一のソフトウェアを対象としており，1 回目から 4 回目のリリースにおける各テスト工程で採取されたフォールト発見数データである [2].

さらに，適合性評価基準として，次式で定義される MSE（平均偏差平方和，mean squared errors）を適用する.

$$\mathrm{MSE} = \frac{1}{K}\sum_{k=1}^{K}\left(y_k - \hat{H}(s_k, u_k)\right)^2. \tag{17}$$

一例として，図 1 に DS2 において推定された習熟 S 字形 2 変量 SRGM の挙動を示す．図 1 より，カレンダ時間および 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

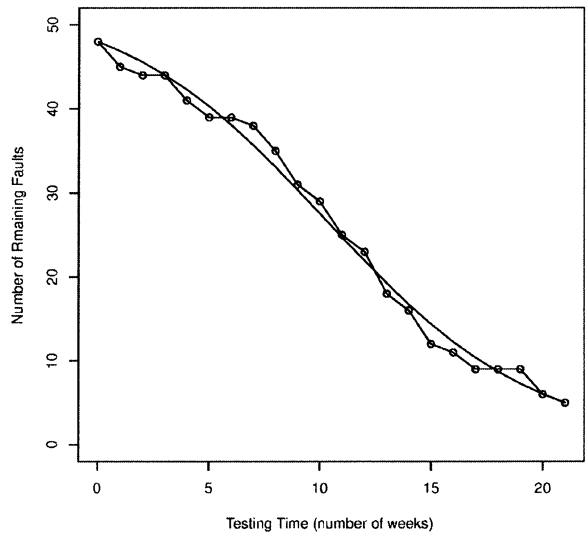

図 2： DS2 において推定された期待残存フォールト数（CES-ISS，$\hat{M}(21, 8736) = 5$）.

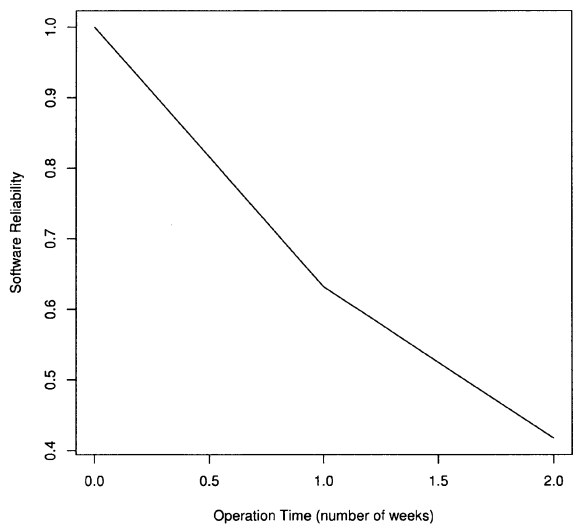

図 3： DS2 において推定されたソフトウェア信頼度関数（CES-ISS，$\hat{R}(2.0|21, 8736) = 0.418$）.

表 1： DS2 における各提案モデルのパラメータ推定の結果.

	$\hat{a}$	$\hat{b}$	$\hat{l}$	$\hat{\alpha}$	$\hat{\rho}$	$\hat{e}$
EXP	317.77	0.0069	-	-	-	-
CES-EXP	101.84	0.024	-	0.99	0.025	1.026
DSS	55.90	0.13	-	-	-	-
CES-DSS	57.89	0.13	-	0.99	-0.108	0.903
ISS	47.99	0.21	0.22	-	-	-
CES-ISS	47.99	0.036	0.10	0.85	0.283	**1.395**

表 2： MSE に基づく適合性比較の結果.

	EXP	CES-EXP	DSS	CES-DSS	ISS	CES-ISS
DS1	221.9953	**205.9784**	**188.9298**	209.1962	96.6514	**96.6090**
DS2	**7.1421**	10.2266	3.4213	**3.3543**	1.7726	**1.7725**
DS3	20.1675	**20.1572**	28.3760	**27.6443**	**11.7432**	13.7962
DS4	31.1986	**26.5182**	14.0370	**13.2427**	**6.5457**	9.9866
DS5	27.2003	**23.5581**	10.9310	**5.8385**	2.3095	**2.1035**
DS6	6.0088	**2.6362**	1.0948	**1.0890**	0.9520	**0.9507**

わかる．次に，図 2 および図 3 に，DS2 において推定された期待残存フォールト数とソフトウェア信頼度関数をそれぞれ示す．リリース後 2 週間目におけるソフトウェア信頼度は $\hat{R}(2.0|21,8736) = 0.418$，期待残存フォールト数は $\hat{M}(21,8736) = 5$ とそれぞれ推定された．

さらに，表 1 に，DS2 における従来モデルおよび提案モデルのパラメータ推定の結果を示す．これらのパラメータは，最尤法を用いて推定した．表 1 より，推定されたパラメータ $\hat{\alpha}$ から，テスト時間要因の方が信頼度成長要因としての影響度合いが大きいことがわかる．また，推定された代替の弾力性 $\hat{e}$ が 1 以上であることから，一定のソフトウェア信頼度成長を維持するために，テスト時間要因とテスト労力要因を代替することは容易であると評価することができる．つまり，これらの信頼度成長要因間で代替を行う場合，比較的長期間，ソフトウェア信頼度成長を維持することが可能であることを意味する．最後に，表 2 に，DS1～DS6 における従来モデルと提案モデルとの適合性比較結果を示す．表 2 より，従来モデルよりも提案モデルの適合性が高いことが確認できる．

6 おわりに

本研究では，CES 型関数を用いて信頼度成長要因としてのテスト時間を表現し，新たな 2 変量 SRGM を提案した．特に，テスト工程におけるカレンダ時間だけでなく，CPU 時間/実行時間といったテスト労力の投入過程を提案モデルに反映し，実際のフォールト発見過程を従来モデルよりも精度良く記述した．さらに，信頼度成長要因間における代替パラメータを導入することで，テスト時間要因とテスト労力要因における代替の弾力性を定量的に評価した．

本研究で適用した CES 型関数は，種々の関数を一般化し，代替の弾力性に関する制約がコブ・ダグラス型関数よりも緩和されている．そのため，CES 型テスト時間関数は，コブ・ダグラス型テスト時間関数よりも多くのテスト工程に適用することができる．しかしながら，3 つ以上の信頼度成長要因を考慮する場合，すべての信頼度成長要因の組み合わせにおいて代替の弾力性が異なり，一定であるような生産関数は存在しない．したがって，今後の課題としては，3 つ以上の信頼度成長要因を

考慮するため，複数の生産関数を組み合わせて構築される組み合わせ型（入れ子型）生産関数を適用した場合について議論する必要がある．さらに，最適リリース（出荷）時刻の推定等，開発管理面への応用問題について適用例を示す必要がある．

謝辞

本研究は，電気通信普及財団 2016 年度研究調査助成（システム技術系分野）の助成を受けたものです．

参考文献

[1] 入谷純，加茂知幸,「経済数学」，東洋経済新報社，2016.

[2] A. Wood, "Predicting software reliability," *IEEE Computer Magazine*, Vol. 11, pp. 69–77, 1996.

[3] B. Anniprincy and S. Sridhar, "Two dimensional software reliability growth models using Cobb-Douglas production function and Yamada S-shaped model," *International Journal of Software Engineering and Simulation*, Vol. 2, Issue. 2, pp. 1–11, 2014.

[4] H. Pham, *Software Reliability*, Springer-Verlag, Singapore, 2000.

[5] 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.

[6] M. Ohba, "Software reliability analysis models," *IBM Journal of Research and Development*, Vol. 28, No. 4, pp. 428–443, 1984.

[7] 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.

[8] 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.

[9] 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.

[10] 井上真二，山田茂，「2 変量ワイブル型ソフトウェア信頼度成長モデルとその適合性評価」，情報処理学会論文誌，Vol. 49，No. 8，pp. 2851–2861，2008 年 5 月．

[11] 井上真二，山田茂，「高信頼性ソフトウェア開発のための最適テスト労力投入問題」，日本信頼性学会誌『信頼性』，Vol. 32，No. 1，pp. 40–46，2010 年 1 月．

[12] S. Yamada, *Software Reliability Modeling —Fundamentals and Applications—*, Springer-Verlag, Tokyo/Heidelberg, 2014.