日本オペレーションズ・リサーチ学会
2005年春季研究発表会

順序統計量に基づく
離散型ソフトウェア信頼度成長モデルに関する一考察

02005295　鳥取大学　*井上 真二　INOUE, Shinji
01702425　鳥取大学　山田 茂　YAMADA, Shigeru

1　はじめに

ソフトウェア信頼度成長モデル（software reliability growth model : SRGM）[7] は，テスト工程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，確率・統計論により記述して，ソフトウェアの信頼性を定量的に計測・評価するためのモデルである．現在までに数多くの SRGM が提案されている一方，順序統計量や無限サーバ待ち行列理論などの概念に基づいて，既存のいくつかの SRGM を特別な場合として包含できるような一般化 SRGM の構築に関する議論も数多くなされている [1, 4, 5]．特に，Langberg and Singpurwalla は，フォールト発見時間データの順序統計量に注目することで，これまで提案されているいくつかの非同次ポアソン過程（nonhomogeneous Poisson process : NHPP）モデルがフォールト発見時間の確率分布により，分類可能であることを議論している [4]．近年，岡村らは，フォールト発見時間の確率分布に関して，指数分布，ガンマ分布，ワイブル分布を特殊の場合として包含できる新たな一般化ガンマ分布とよばれる確率分布関数を構築して，一般化順序統計量の考え方に基づいた一般化 SRGM の構築およびそのパラメータ推定手法について議論している [5]．このような一般化 SRGM は，実際のテスト工程において観測および分析された様々なソフトウェア故障発生パターンに直面しても，そのパターンを容易にモデルへと反映できる特徴をもつ．

しかしながら，これまでの一般化 SRGM に関する議論は，連続時間モデルについての議論が大半である．実際のテスト工程における信頼性データの離散的な記録作業や，消化されたテスト項目数やテストラン試行回数などの離散的な時間の経過に伴うソフトウェア信頼度成長過程を記述する場合に着目すると，離散時間モデルが整合性を有したモデルであると考えられる．このような背景に基づいて，近年では，累積ベルヌーイ過程に基づいた離散型 NHPP モデルの一般化手法が提案されている [6]．本研究では，Langberg and Singpurwalla が提案した一般化順序統計量の枠組みにおいて，フォールト発見時間の確率分布が離散型確率分布に従う場合について考え，離散型 SRGM の一般化に関する様々な考察を行う．

2　一般化順序統計量モデル

フォールト発見時間の確率分布が離散時間分布に従う場合に対する一般化順序統計量モデルを構築するにあたり，以下の仮定を設ける．

(A1)　テスト工程において，ソフトウェア故障が観測された場合，その原因となるソフトウェアフォールトは直ちに修正・除去される．

(A2)　テスト開始前にソフトウェア内に潜在する総フォールト数（初期潜在フォールト数）N_0 は，ある確率分布に従う確率変数とする．

(A3)　各ソフトウェア故障は，それぞれ独立かつ時間に関してランダムに発生し，各ソフトウェア故障発生時刻は，それぞれ離散型確率分布 $P(i)$ に従う．

ここで，(A3) は，ソフトウェア故障発生時刻を表す離散型確率変数列を $\{X_t, t=1,\cdots,N_0\}$ と定義したとき，X_t が互いに独立かつ同一の離散型確率分布 $P(i)$ に従うことを意味している．

まず，$\{N(i), i=0,1,\cdots\}$ をテスト開始後 i 期目までに発見される総フォールト数を表す離散型確率過程であるとする．これより，テスト開始前においてソフトウェア内に n 個のフォールトが存在しているという条件の下で，テスト開始後 i 期目までに m 個のフォールトが発見される確率は，

$$\Pr\{N(i)=m \mid N_0=n\} = \binom{n}{m}\{P(i)\}^m\{1-P(i)\}^{n-m}, \tag{1}$$

と求められる．式 (1) より，テスト開始後 i 期目までに発見される総フォールト数の確率関数は，

$$\Pr\{N(i)=m\} = \sum_n \binom{n}{m}\{P(i)\}^m\{1-P(i)\}^{n-m}\Pr\{N_0=n\}, \tag{2}$$

のように表現される．また，式 (2) より，代表的なソフトウェア信頼性評価尺度の 1 つであるソフトウェア信頼度関数が導出できる．ソフトウェア信頼度とは，テスト開始後 i 期目までテストが進行しているとき，時間区間 $(i, i+h](i,h=0,1,\cdots)$ においてソフトウェア故障が発生しない確率として定義される．したがって，ソフトウェア信頼度関数 $R(i,h)$ は，

$$\begin{aligned} R(i,h) &= \sum_k \Pr\{N(i+h)=k \mid N(i)=k\} \cdot \Pr\{N(i)=k\} \\ &= \sum_k \Big[\{P(i)\}^k\{1-P(i+h)\}^{-k} \sum_n \binom{n}{m}\{1-P(i+h)\}^n\cdot\Pr\{N_0=n\}\Big], \end{aligned} \tag{3}$$

と求められる．

式 (2) および式 (3) は，初期潜在フォールト数を表す確率変数 N_0 に対して，ある適切な確率分布を与えることで，テスト工程におけるフォールト発見事象およびソフトウェア信頼度の時間的挙動は，それぞれ具体的に特徴付けられる．

3 初期潜在フォールト数の確率分布に基づく考察

本研究では，式 (2) および式 (3) における初期潜在フォールト数 N_0 の確率分布が，パラメータ α のポアソン分布もしくはパラメータ K および λ の二項分布に従う場合について，それぞれ考察を与える．

3.1 初期潜在フォールト数がポアソン分布に従う場合

初期潜在フォールト数 N_0 が，次のようなパラメータ α のポアソン分布に従う場合を考える．

$$\Pr\{N_0 = n\} = \frac{\alpha^n}{n!}\exp(-\alpha) \qquad (\alpha > 0). \tag{4}$$

式 (2) および式 (4) より，テスト開始後 i 期目までに発見される総フォールト数の確率関数は，次式のように求められる．

$$\begin{aligned}&\Pr\{N_P(i) = m\}\\ &\qquad = \frac{\{\alpha P(i)\}^m}{m!}\exp\{-\alpha P(i)\}.\end{aligned} \tag{5}$$

また，式 (3) より，ソフトウェア信頼度関数 $R_P(i,h)$ は，

$$R_P(i,h) = \exp[-\alpha\{P(i+h) - P(i)\}], \tag{6}$$

と求められる．

式 (5) は，平均値関数が $\alpha P(i)$ である離散型 NHPP[8] と等価となり，各ソフトウェア故障発生時刻に対する離散型確率分布を適切にあてはめることにより，離散型 NHPP モデルを構築することができる．すなわち，式 (5) は，離散型 NHPP モデルを一般的に表現しているものと言える．例えば，各ソフトウェア故障発生時刻がパラメータ p の幾何分布に従う場合，離散型 NHPP の平均値関数 $H(i)$ は，

$$\begin{aligned}H(i) &\equiv \alpha P(i) = \alpha\{1-(1-p)^i\}\\ &\qquad\qquad (0 < p < 1),\end{aligned} \tag{7}$$

と求められ，既存の離散型指数形 SRGM[2] と本質的に等価となる．

3.2 初期潜在フォールト数が二項分布に従う場合

さらに本研究では，初期潜在フォールト数 N_0 が，次のようなパラメータ K および λ をもつ二項分布に従う場合を考える．

$$\begin{aligned}&\Pr\{N_0 = n\} = \binom{K}{n}\lambda^n(1-\lambda)^{K-n}\\ &\qquad\qquad (0 < \lambda < 1\,;\, n = 0,1,\cdots,K).\end{aligned} \tag{8}$$

ここで，式 (8) は，次のような物理的意味 [3] をもっている．

(a) テスト開始時点におけるプログラムは，K コード行 (LOC) で構成される．

(b) 各コードは，それぞれ一定の確率 λ で 1 個のフォールトを含む．

(c) プログラム内のコード中に潜在するフォールトにより引き起こされるソフトウェア故障は，それぞれ独立かつランダムに発生する．

式 (2) および式 (8) より，テスト開始後 i 期目までに発見される総フォールト数の確率関数は，

$$\begin{aligned}&\Pr\{N_B(i) = n\}\\ &\qquad = \binom{K}{n}\{\lambda P(i)\}^m\{1-\lambda P(i)\}^{K-m},\end{aligned} \tag{9}$$

と求められ，

$$\mathrm{E}[N_B(i)] = K\lambda P(i), \tag{10}$$

$$\mathrm{Var}[N_B(i)] = K\lambda P(i)\{1-\lambda P(i)\}, \tag{11}$$

をもつ二項過程に従うことがわかる．ここで，E[·] および Var[·] は，それぞれ期待値および分散を表す．また，式 (3) より，ソフトウェア信頼度関数 $R_B(i,h)$ は，

$$R_B(i,h) = [1 - \lambda\{P(i+h) - P(i)\}]^K, \tag{12}$$

と求められる．

謝辞

本研究の一部は，日本学術振興会科学研究費補助金基盤研究 (C)(2)（課題番号 15510129）および笹川科学研究助成（研究番号 16-064）の援助を受けたことを付記する．

参考文献

[1] 土肥 正，松岡 寿明，尾崎 俊治，“ソフトウェア信頼性評価のための無限サーバ待ち行列モデル，” 電子情報通信学会論文誌，Vol. J83-A, No. 5, pp. 536–544, 2000 年．

[2] S. Inoue and S. Yamada, “Discrete software reliability assessment with discretized NHPP models,” *Computers & Mathematics with Applications: An International Journal*, to be published.

[3] M. Kimura, S. Yamada, H. Tanaka, and S. Osaki, “Software reliability measurement with prior-information on initial fault content,” *Trans. IPS Japan*, Vol. 34, No. 7, pp. 1601–1609, 1993.

[4] N. Langberg and N.D. Singpurwalla, “A unification of some software reliability models,” *SIAM J. Stat. Comput.*, Vol. 6, No. 3, pp. 781–790, 1985.

[5] 岡村 寛之，安藤 光昭，土肥 正，“一般化ガンマソフトウェア信頼性モデル，” 電子情報通信学会論文誌，Vol. J87-D-1, No. 8, pp. 805–814, 2004 年．

[6] 岡村 寛之，村山 篤史，土肥 正，“累積ベルヌーイ過程による離散型ソフトウェア信頼性モデルの統一化とパラメータ推定手法に関する考察，” 2002 年日本オペレーションズ・リサーチ学会春季研究発表会アブストラクト集，pp. 102–103.

[7] 山田 茂，藤原 隆次，「ソフトウェアの信頼性：モデル，ツール，マネジメント」，プロジェクトマネジメント学会（PM 学会教育・出版シリーズ (1)），2004 年．

[8] S. Yamada and S. Osaki, “Discrete software reliability growth models,” *Appli. Stoc. Models and Data Ana.*, Vol. 1, No. 1, pp. 65–77, 1985.