

## マルチスケール解析によるソフトウェア強度関数推定法の 信頼性評価への応用

広島大学大学院工学研究科 肖 霄, 土肥 正

Xiao Xiao and Tadashi Dohi  
Graduate School of Engineering  
Hiroshima University, Japan

### 1 はじめに

ソフトウェアの信頼性を評価するためには, ソフトウェア強度関数を観測データから高精度で推定する手法が必要とされる. ソフトウェア信頼性モデル (Software Reliability Model; SRM) は, テスト段階で観測されたフォールト検出に関する各種統計情報に基づいて, ソフトウェア障害が発生する現象を記述する数学モデルであり, ソフトウェア信頼性を定量的に評価するために利用される. 一般に, ソフトウェアフォールト検出過程を非同次ポアソン過程 (Non-Homogeneous Poisson Process; NHPP) に従うと仮定することが多く, NHPP に基づいた SRM はソフトウェアの信頼性評価を実施する上で最も広範に適用されている [3]. これまでに提案された NHPP に基づいた SRM の多くはパラメトリックモデルであった [5]. パラメトリックモデルでは期待累積フォールト数を表す平均値関数に滑らかな関数形を仮定するため, 日々のテストにおいて変動する検出フォールト数の詳細な振舞いを原理的に記述できないことが弱点として挙げられる. よって, モデリングにおけるソフトウェアデバッグに関するシナリオを必要としないノンパラメトリックモデルの重要性が増してきている.

近年, 計数過程のノンパラメトリック推定を行うウェーブレット縮小推定 (Wavelet Shrinkage Estimation; WSE) が急速に整備されてきた [1]. 先行研究 (Xiao and Dohi [6]) では WSE を NHPP に基づいた SRM のノンパラメトリック推定法として初めてソフトウェア信頼性分野に導入し, ソフトウェア信頼性評価における WSE の有効性について検証した. Xiao and Dohi [6] は特に, データ変換を伴う WSE に着目し, 従来のパラメトリック推定法である最尤法 (Maximum Likelihood Estimation; MLE) や最小二乗法 (Least Squares Estimation; LSE) に基づいた評価よりも単位テスト時間当たりの検出フォールト数を高い精度で推定できることを示した. さらに, ポアソン過程の強度関数を WSE で推定する際にデータ変換が原因で *oversmoothing* などの問題点が生じやすいという課題を解決するために, Xiao and Dohi [8] では, TIPSH (Translation-Invariant Poisson Smoothing using Haar Wavelets) アルゴリズム (文献 [2]) を実際のテスト工程で観測されたソフトウェアフォールトデータの解析に適用し, データ変換を行わない WSE について考察するとともに, 文献 [6] における評価結果との比較を行った. 説明の便宜上, 2つの手法を区別してデータ変換を伴う WSE を DT-WSE, データ変換を伴わない WSE を NDT-WSE によって表記する.

Timmermann and Nowak [4] はマルチスケール解析 (Multiscale Analysis) とベイズ推定を融合したベイジアンマルチスケール推定 (Bayesian Multiscale Estimation; BME) を開発し, ポアソン強度関数の推定を行った. 本稿では, BME に着目し, そのソフトウェア信頼性評価への応用可能性について考察する. 数値例では, 従来の DT-WSE や NDT-WSE と比較しながら, 両者の推定性能を適合性の観点から評価する.

## 2 離散型 NHPP モデル

ここでは単位テスト時間当たりの検出フォールト数の振舞いに着目するため、一般性を失うことなく、離散型 NHPP モデルについて述べる [7].

今、時間区間を適当なスケールで離散化した時刻列  $k = 0, 1, 2, \dots$  を考える. テスト時刻  $k$  において観測されるソフトウェアフォールト数を  $Y_k$ , その累積値を  $N_k = \sum_{k=0}^k Y_k$  で表し,  $Y_0 = N_0 = 0$  とする. このとき, 確率過程  $\{N_k : k = 0, 1, 2, \dots\}$  が離散型 NHPP であるとは, その確率関数が

$$\Pr\{N_k = m\} = \frac{\{\Lambda_k\}^m}{m!} \exp\{-\Lambda_k\}, \quad m = 0, 1, 2, \dots \quad (1)$$

によって与えられることをいう. ここで,  $E[N_k] = \Lambda_k$  は離散型 NHPP の平均値関数と呼ばれ, 時刻  $k$  までに検出される期待累積フォールト数を表す. また, 強度関数  $\lambda_k = \Lambda_k - \Lambda_{k-1}$  は  $k (\geq 1)$  日目のテストにおいて新たに検出された期待フォールト数であり,  $E[Y_k] = \lambda_k$  が成立つ. これより, 離散型 NHPP モデルは平均値関数 (強度関数) によって特徴付けられ, 離散関数  $\Lambda_k$  ( $\lambda_k$ ) の形状によって多様なフォールト検出パターンを表現することが可能である.

ソフトウェア強度関数を求める問題は NHPP 強度関数の推定に帰着されるため, 以下では強度関数  $\lambda_k$  の推定法について紹介する. マルチスケール推定を適用するために, 時間区間  $[0, T]$  を  $n (= 2^J; J = 1, 2, \dots)$  等分することを考える. 説明の便宜上,  $y_k = x_k - x_{k-1}$  ( $k = 1, 2, \dots, n$ ) は強度  $\lambda_k$  を持つポアソン確率変数  $Y_k$  の実現値とする. 即ち,

$$Y_k = \lambda_k + \eta_k, \quad k = 1, 2, \dots, n. \quad (2)$$

ただし,  $\eta_k$  はポアソン型白色雑音を表す確率変数であり,  $x_k$  ( $k = 0, 1, 2, \dots, n$ ) は各時間区間までに検出される累積フォールト数を表す.

## 3 マルチスケール解析とウェーブレット解析

マルチスケール解析とは, その名の通り, 異なるスケールの解析を組み合わせるものであり, 画像解析等ではよく使われる解析方法の一つである. データを様々なレベルに落として, ミクロな特徴を表す fine level の情報からマクロな特徴を表す coarse level を推定し, データの背後にある真の関数を求めることができる.

マルチスケール解析の最も基本的な手法として知られているのは, ウェーブレット解析でもよく用いられるハールウェーブレット変換 (Haar Wavelet Transform; HWT) である. 従来のウェーブレット縮小推定 (DT-WSE, NDT-WSE) もベイジアンマルチスケール推定 (BME) もマルチスケール解析の枠組み内にあるものであり, 両者とも HWT をベースにした推定手法である. HWT により, 観測データから滑らかさ (smoothness) と細かさ (detailness) に関する情報が fine level から coarse level まで, レベルごとに抽出される. WSE は細かさに関する情報のみに着目し, データに含まれるノイズを除去することで推定するが, BME は全ての情報を利用することやノイズ除去を行わないといった点で WSE と異なる. BME では, fine level に落とされたデータに事前密度を与えて推定を行うため, 推定精度の向上につながる手法であると期待される.

また, マルチスケール解析をソフトウェア信頼性評価に導入した理由として, 推定精度の向上だけではない. NHPP モデルでは各微小区間におけるフォールト数を表わす確率変数  $Y_k$  はポアソン分布に従うため, 隣り合うデータを足したり引いたりしても, その和や差はまたポアソン分布に従うという性質は変わらない. 従って, データをレベルごとに処理して推定精度を高められることや, 確率的な性質を全てのレベルにおいて保つことができることから, 本稿ではマルチスケール解析に注目した. これらの手法を用いて, NHPP 強度関数の推定について考える.

### 3.1 ハールウェーブレット変換 (HWT)

本稿で使われる HWT に関連する記号をベクトル表現で定義する.

観測データ  $c = c_0 = (c_{0,0}, c_{0,1}, \dots, c_{0,N-1})$  から強度関数  $\lambda = \lambda_0 = (\lambda_{0,0}, \lambda_{0,1}, \dots, \lambda_{0,N-1})$  を推定する問題を考える.  $c_0$  と  $\lambda_0$  の添え字 0 は解像度レベル (resolution level) を表す. ここで,  $N = 2^J$  は観測データの長さであり,  $J$  は任意の整数である.  $\lambda_{0,k}$  ( $k = 0, 1, \dots, N-1$ ) は時刻  $k$  におけるフォールト発見数を表しており,  $c_{0,k}$  はその実現値である. すなわち,

$$c_0 \sim \text{Poisson}(\lambda_0). \quad (3)$$

HWT により, 解析対象関数  $\lambda$  はスケーリング係数  $\lambda_{j,k}$  とウェーブレット係数  $\theta_{j,k}$

$$\lambda_{j,k} = \lambda_{j-1,2k} + \lambda_{j-1,2k+1}, \quad (4)$$

$$\theta_{j,k} = \lambda_{j-1,2k} - \lambda_{j-1,2k+1} \quad (5)$$

に展開される. ただし,  $j = 1, 2, \dots, J$ ,  $k = 0, 1, \dots, N/2^j - 1$ . 同様, 観測データ  $c$  のスケーリング係数  $c_{j,k}$  とウェーブレット係数  $d_{j,k}$  はそれぞれ

$$c_{j,k} = c_{j-1,2k} + c_{j-1,2k+1}, \quad (6)$$

$$d_{j,k} = c_{j-1,2k} - c_{j-1,2k+1} \quad (7)$$

のように定義される.

### 3.2 Multiscale Multiplicative Innovations Model

従来のウェーブレット縮小推定ではウェーブレット係数  $d_{j,k}$  に含まれるノイズを除去することで強度関数を推定する. ベイジアンマルチスケール推定ではノイズ除去を行わず, その代わりに事前密度モデルを利用して強度関数の推定を行う. Timmermann and Nowak [4] が提案した MMI モデル (Multiscale Multiplicative Innovations Model) について紹介する.

スケーリング係数  $\lambda_{j,k}$  とウェーブレット係数  $\theta_{j,k}$  の確率変数をそれぞれ  $\Lambda_{j,k}$ ,  $\Theta_{j,k}$  で表わし, 摂動変数 (perturbation variable)  $\Delta_{j,k}$  を導入して両者の間に次の関係式が成り立つと仮定する.

$$\Theta_{j,k} = \Lambda_{j,k} \Delta_{j,k}. \quad (8)$$

このように, 全てのウェーブレット係数はそれに対応するスケーリング係数の独立な摂動として捉えることができる. 式 (4) と (5) をこれらの係数に適応して,

$$\Lambda_{j-1,k} = \frac{1}{2}(\Lambda_{j,[k/2]} + (-1)^k \Theta_{j,[k/2]}) \quad (9)$$

を得ることができる.  $[\cdot]$  は中身を超えない最大の整数を意味する. 今, 式 (8) を式 (9) に代入し, さらに確率変数  $Y_{j,k} = \frac{1}{2}(1 + \Delta_{j,k})$  を導入すると,

$$\Lambda_{j-1,2k} = \Lambda_{j,k} Y_{j,k}, \quad (10)$$

$$\Lambda_{j-1,2k+1} = \Lambda_{j,k}(1 - Y_{j,k}) \quad (11)$$

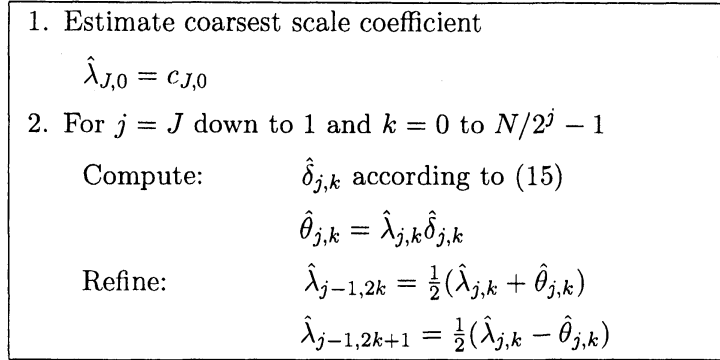


図 1: Bayesian Multiscale Estimation

が得られる。積を使って異なるスケール間のスケーリング係数の関係を表すことができることから、文献 [4] では MMI モデルと名づけた。

MMI モデルでは摂動変数  $\Delta_{j,k}$  の事前密度を与えることによってウェーブレット係数やスケーリング係数の推定を行う。本稿では、文献 [4] と同じくベータ混合密度関数 (beta-mixture densities)

$$f(\delta) = \sum_{i=1}^M p_i \frac{(1-\delta^2)^{s_i-1}}{B(s_i, s_i) 2^{2s_i-1}}, \quad (-1 \leq \delta \leq 1) \quad (12)$$

を利用する。ただし、 $B(u, v) = \int_0^1 t^{u-1} (1-t)^{v-1} dt$  はオイラーベータ関数であり、 $p_i$  は  $i$  番目のベータ密度  $\frac{(1-\delta^2)^{s_i-1}}{B(s_i, s_i) 2^{2s_i-1}}$ , ( $s_i \geq 1$ ) の重みで  $\sum_{i=1}^M p_i = 1$  を満たす。パラメータ  $p_i, s_i$  はモーメントマッチング法によって推定した。ベータ混合密度関数を用いた理由として、(i)  $[-1, 1]$  においてサポートを持つこと、(ii) 原点に関して対称であること、(iii) 単峰性であることが挙げられる。

### 3.3 BME アルゴリズム

ベイズの定理を式 (4), (5), (8), (12) に適応して式変換を施すと、ウェーブレット係数、スケーリング係数、摂動変数の事後密度はそれぞれ、

$$\hat{\theta}_{j,k} = \hat{\lambda}_{j,k} \hat{\delta}_{j,k}, \quad (13)$$

$$\hat{\lambda}_{j,k} = \frac{1}{2}(\hat{\lambda}_{j+1, [k/2]} + (-1)^k \hat{\theta}_{j+1, [k/2]}), \quad (14)$$

$$\hat{\delta}_{j,k} = d_{j,k} \frac{\sum_{i=1}^M p_i \frac{B(s_i + c_{j-1, 2k}, s_i + c_{j-1, 2k+1})}{B(s_i, s_i) (2s_i + c_{j,k})}}{\sum_{i=1}^M p_i \frac{B(s_i + c_{j-1, 2k}, s_i + c_{j-1, 2k+1})}{B(s_i, s_i)}} \quad (15)$$

のように導かれる。これらをもとにした BME アルゴリズムを図 1 に示す。

## 4 数値例

ここでは文献 [6] と同じソフトウェアフォールトデータを用いて、離散型 NHPP モデルにおけるパラメータ推定法の比較を行う。使用したデータは各テスト日までに検出されるソフトウェアフォールト数を記録されたもので、(テスト終了時刻, 総累積フォールト数) はそれぞれ (62, 136),

表 1: 代表的な離散型 NHPP モデル.

モデル名	平均値関数 $\Lambda_k$ ( $k = 1, 2, 3, \dots$ )
幾何分布 (GE)	$\omega\{1 - (1 - p)^k\}$
負の二項分布 (NB)	$\omega\{\sum_{i=1}^k \frac{(r+i-1)!}{(r-1)!i!} p^r (1-p)^{i-1}\}$
離散ワイブル分布 (DW)	$\omega\{1 - p^{k^r}\}$

(注:  $\Lambda_0 = 0$ )

(41, 351), (73, 367), (81, 461) である. 各パラメータ推定法に基づいて算出された平均値関数  $\Lambda_k$  と強度関数  $\lambda_k$  を用いて, 実データとの平均二乗誤差 (Mean Square Error; MSE)

$$\text{MSE}_1 = \frac{\sqrt{\sum_{k=1}^n (\Lambda_k - x_k)^2}}{n}, \quad (16)$$

$$\text{MSE}_2 = \frac{\sqrt{\sum_{k=1}^n (\lambda_k - y_k)^2}}{n} \quad (17)$$

と最大対数尤度 (Maximum Log Likelihood; MLL) を比較し, 誤差が小さくなるか, または最大対数尤度が大きくなるパラメータ推定法を最良の方法とする.

#### 4.1 パラメトリック手法との比較

まず文献 [6] と同じように, 表 1 に示す代表的な離散型 NHPP モデルを扱い, パラメトリック推定法である MLE 及び LSE による推定結果との比較を行う. その結果を表 2 に示す.  $\text{MSE}_1$ ,  $\text{MSE}_2$ , MLL のいずれの評価基準を見ても, BME は MLE より優れた推定法であることが結論付けられる. 特に, BME では尤度を最大化する操作を行っていないにもかかわらず, MLE より大きい対数尤度を与えることが可能である. ほぼ同様の結果は LSE との比較結果にも見受けられる. パラメトリックなモデルに LSE を適用したとしても, BME の推定精度を超えることは非常に稀であり, 離散型 NHPP モデルの平均値関数と実データの距離を最小にするという目的に対しても, BME に基づいた方法は非常に優れていると結論付けることができる. 図 2 は DS1 における推定結果を視覚的に表したものである. 累積値の比較では 3 つの推定法の違いに大きな差はないように見受けられるが, 強度に着目した比較を行うとパラメトリックモデルは関数形を限定するため滑らかな曲線になっているのに対し, BME では実データの振動に追従した推定結果を得ており, 各テスト日当たりのフォールト検出数の変動を高精度で捉えていることがわかる. 換言すれば, 最尤法並びに最小二乗法に基づいたパラメトリックモデルは累積フォールト数の挙動を大まかに捉えることが可能であるが, 各テスト日ごとに検出されるフォールト数のようにマイクロな挙動を把握することができず, これが従来法の弱点であるとも言える.

#### 4.2 ウェーブレット手法との比較

次に, Xiao and Dohi [6] が提案した DT-WSE 及び Xiao and Dohi [8] が提案した NDT-WSE との比較結果を表 3 に示す.  $\text{HAT}(\cdot, \cdot)$ ,  $\text{HFT}(\cdot, \cdot)$  はそれぞれ Anscombe Transform と Fisz Transform を利用した DT-WSE を表しており,  $\text{H}(\cdot, \cdot)$  は NDT-WSE,  $\text{HTI}(\cdot, \cdot)$  は TIPSH アルゴリズムを使用した NDT-WSE を意味し, s, ut, ldt はそれぞれ soft thresholding, universal threshold, level-dependent threshold の略である.

表 2: パラメトリック手法との比較.

DS1	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL	DS2	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	0.49	0.28	-128.07	BME	1.71	0.58	-107.19
MLE(GE)	0.68	0.31	-116.75	MLE(GE)	3.45	1.02	-157.86
MLE(NB)	1.05	0.32	-112.65	MLE(NB)	2.51	1.00	-141.50
MLE(DW)	0.69	0.31	-116.69	MLE(DW)	2.14	0.94	-135.79
LSE(GE)	0.59	0.31	-142.73	LSE(GE)	2.83	1.01	-165.77
LSE(NB)	0.92	0.32	-137.90	LSE(NB)	2.33	0.99	-145.78
LSE(DW)	0.59	0.31	-142.63	LSE(DW)	1.97	0.93	-140.66

DS3	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL	DS4	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	0.69	0.26	-158.73	BME	3.28	0.39	-204.34
MLE(GE)	0.81	0.30	-166.43	MLE(GE)	1.19	0.50	-237.94
MLE(NB)	1.92	0.32	-169.03	MLE(NB)	2.35	0.56	-261.66
MLE(DW)	0.96	0.29	-162.86	MLE(DW)	1.03	0.49	-235.15
LSE(GE)	0.80	0.30	-170.62	LSE(GE)	1.11	0.50	-248.49
LSE(NB)	1.73	0.32	-174.77	LSE(NB)	2.16	0.56	-275.27
LSE(DW)	0.78	0.30	-169.00	LSE(DW)	1.02	0.49	-245.56

評価基準のいずれを見ても、BME は WSE に基づいた結果より小さい誤差と大きい対数尤度を提供することが分かる。これにより、事前情報を利用したベイジアンアプローチはソフトウェア信頼性評価へ適用可能であり、推定精度に関して従来のウェーブレット縮小推定を上回る手法であると結論づけることができる。なお、WSE も BME もノンパラメトリック推定手法であるため、推定結果に差が小さく両者のプロットはほぼ重なる。従って、ここでは数値的な結果のみを示すことにした。

## 5 まとめと今後の課題

本研究では、NHPP 強度関数の推定方法として、ベイズ統計の考え方を導入したマルチスケール解析法に着目し、ソフトウェア信頼性評価への適用可能性について検討した。実データを用いた検証実験において、先行研究によるウェーブレット縮小推定に基づいた評価よりも、BME は高い推定精度を有することが示した。今後は、より多くのデータセットを用いて本研究で得られた知見を確かめる必要がある。また、事前密度関数に含まれるパラメータのチューニングなどが課題として挙げられる。

## 参考文献

- [1] P. Besbeas, I. De Feis and T. Sapatinas, "A comparative simulation study of wavelet shrinkage estimators for Poisson counts," *International Statistical Review*, **72** (2), pp. 209–237, 2004.

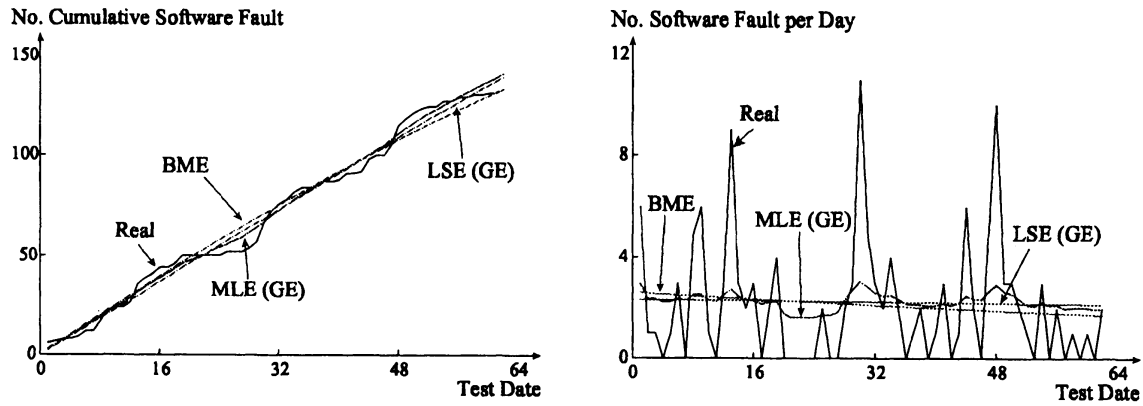


図 2: DS1 における推定値の振舞い.

表 3: ウェーブレット手法との比較.

DS1	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	0.49	0.28	-128.07
HAT(s, ut)	2.57	0.32	-145.39
HFT(s, ut)	0.59	0.31	-142.65
H(s, ldt)	0.39	0.26	-120.49
HTI(s, ldt)	0.44	0.26	-119.87

DS3	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	0.69	0.26	-158.73
HAT(s, ut)	1.82	0.30	-171.50
HFT(s, ut)	1.01	0.30	-171.74
H(s, ldt)	0.75	0.26	-159.16
HTI(s, ldt)	0.70	0.27	-160.88

DS2	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	1.71	0.58	-107.19
HAT(s, ut)	11.7	1.14	-196.95
HFT(s, ut)	7.23	1.13	-197.13
H(s, ldt)	0.83	0.50	-103.32
HTI(s, ldt)	2.10	0.48	-102.65

DS4	MSE <sub>1</sub>	MSE <sub>2</sub>	MLL
BME	3.28	0.39	-204.34
HAT(s, ut)	13.0	0.60	-310.00
HFT(s, ut)	8.56	0.59	-306.70
H(s, ldt)	1.39	0.37	-201.61
HTI(s, ldt)	2.82	0.37	-206.82

- [2] E. D. Kolaczyk, "Non-parametric estimation of gamma-ray burst intensities using Haar wavelets," *The Astrophysical Journal*, **383** (1), pp. 340–349, 1997.
- [3] M. R. Lyu (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, New York, 1996.
- [4] K. E. Timmermann and R. D. Nowak, "Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging," *IEEE Transactions on Information Theory*, **45** (3), pp. 846–862, 1999.
- [5] X. Xiao and T. Dohi, "On equilibrium distribution properties in software reliability modeling," *Proceedings of 4th International Conference on Availability, Reliability and Security (ARES'09)*, pp. 158–165, IEEE CS Press, 2009.

- [6] X. Xiao and T. Dohi, "Wavelet-based approach for estimating software reliability," *Proceedings of 20th International Symposium on Software Reliability Engineering (ISSRE'09)*, pp. 11–20, IEEE CS Press, 2009.
- [7] S. Yamada and S. Osaki, "Discrete software reliability growth models," *Applied Stochastic Models and Data Analysis*, 1 (1), pp. 65–77, 1985.
- [8] 肖 霄, 土肥 正, "TIPSH アルゴリズムに基づいたウェーブレット縮小推定によるソフトウェア信頼性評価," 京都大学数理解析研究所研究集会報告集-不確実・不確定性下での意思決定過程, pp. 201–208, 2010.