



Title	NHPPに基づくソフトウェア信頼度成長モデルの定量的比較
Author(s)	松本, 健一; 山本, 浩司; 井上, 克郎 他
Citation	情報処理学会研究報告. ソフトウェア工学研究会報告. 1988, 1987-SE-058(7), p. 89-96
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/50182">https://hdl.handle.net/11094/50182</a>
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

NHPPに基づくソフトウェア信頼度成長モデル  
の定量的比較

松本 健一 山本 浩司\* 井上 克郎 菊野 亨 鳥居 宏次

大阪大学 基礎工学部

ソフトウェアのテスト段階から収集したデータに基づいて、ソフトウェアの信頼性を定量的に分析、評価するためのモデルが数多く提案されている。本稿では、学生実験から収集したデータを利用して、非同次ポアソン過程 (NHPP) の理論を応用したソフトウェア信頼度成長モデル (SRGM) の比較を行う。先ず、推定値の誤差評価及び信頼度成長曲線の適合度評価に関し、S字型SRGMが指数型SRGMより優れていることを示す。次に、超指数型SRGMに対し、推定精度を高める新しい方法を提案する。この方法はプログラムの各クラスタのテスト開始時刻が異なることに注目したものである。

## Experimental Comparisons of Software Reliability Growth Models

Ken'ichi MATSUMOTO, Koji YAMAMOTO \*, Katsuro INOUE, Tohru KIKUNO, and Koji TORII

Faculty of Engineering Science, Osaka University  
1-1 Machikaneyama, Toyonaka, Osaka 560, Japan

This paper presents an experimental evaluation of software reliability growth models (SRGM's) described by nonhomogeneous Poisson processes (NHPP's). The experimental data sets are collected from university students' projects, in which each student develops a compiler for a subset of Pascal. First, it will be shown that the S-shaped SRGM is superior to the exponential SRGM in both the accuracy of estimation and goodness of fit. Then, for the hyperexponential SRGM, a new estimation method is proposed to improve the goodness of fit significantly. This new method takes notice of the origin of the growth curve for the hyperexponential SRGM.

---

\*現在、関西電力株式会社

## 1. まえがき

ソフトウェアのテスト段階から収集したデータに基づいて、ソフトウェアの信頼性を定量的に分析、評価するためのモデルが数多く提案されている<sup>[1-4, 6, 9, 11]</sup>。本稿では、いわゆる非同次ポアソン過程 (NHPP) の理論を応用したソフトウェア信頼性のモデルについて議論する。

ソフトウェア信頼度成長モデル (SRGM) では、ソフトウェアのテスト段階において発見されるフォールト (欠陥) の累積数を時間の関数  $H(t)$  として表す。関数  $H(t)$  を二次元平面上に描いたときの曲線の形状によって、SRGMは指数型とS字型に分類される。指数型には指数型 (Exponential) 信頼度成長モデル<sup>[2]</sup>と超指数型 (Hyperexponential) 信頼度成長モデル<sup>[6, 13]</sup>、S字型には遅延S字型 (Delayed S-shaped) 信頼度成長モデル<sup>[11]</sup>と習熟S字型 (Inflection S-shaped) 信頼度成長モデル<sup>[3, 6, 8]</sup>がある。

これらのモデルを比較評価した報告としては文献<sup>[7, 8, 12]</sup>がある。しかし、通常、1つあるいは複数のプロジェクトに対する1組の収集データに基づいて比較評価が行われる。しかも、データ収集の基本単位が日、あるいは月となっている。

本稿では、大阪大学基礎工学部情報工学科で実施した学生実験から収集したフォールトデータに基づいて、上述の4つのSRGMの比較を行う。なお、本比較評価は複数 (ここでは5) 組の、しかも、分単位の収集データに基づいている。

得られた主な結果を次にまとめる。

- (1) 初期フォールト数の推定値と実測値の間の誤差評価、および、信頼度成長曲線の適合度評価に関しては、S字型モデルが指数型モデルより優れていることが分かった。
- (2) テストの初期段階のデータのみ限定してモデルを適用した場合、遅延S字型モデル以外は、初期フォールト数の推定値が実測値よりかなり小さくなった。
- (3) 超指数型信頼度成長モデルに対し、推定の精度を高める方法を提案した。この方法はプログラムのクラスタ毎にテスト開始時刻が異なること、及び、データ収集の基本単位が小さいことに注目したものである。

## 2. ソフトウェア信頼度成長モデル

プログラムテストの実行で、プログラム中のフォールトは発見され、取り除かれる。今、フォールトを取り除く過程で新たなフォールトがプログラム中に作り込まれないと仮定すると、プログラムテストの進行に伴ってプログラムの信頼度は高くなる。

プログラムの信頼度に関するこの現象は、いわゆるソフトウェア信頼度成長モデル (SRGM) を利用すると説明がつく。SRGMでは、テスト時間と発見されるフォールトの累積数の間の関係を表現する信頼度成長関数  $H(t)$  (以降では、信頼度成長曲線とも呼ぶ) を用いてソフトウェアの信頼度を定義する。

SRGMは信頼度成長曲線の形状によって、指数型SRGMとS字型SRGMに分類される (図1参照)。代表的な指数型SRGMには、GoelとOkumoto<sup>[2]</sup>が提案した E(Expo-

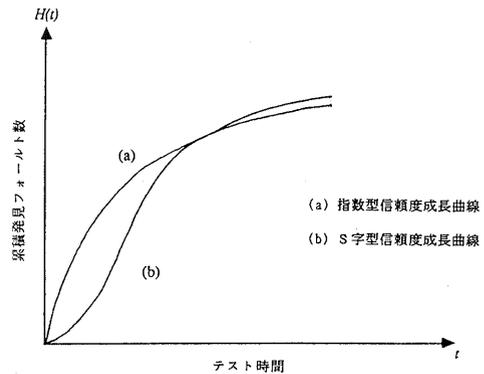


図1. ソフトウェア信頼度成長曲線

表1. 各モデルにおける信頼度成長曲線  $H(t)$  の定義

		信頼度成長曲線 $H(t)$	パラメータ
指数型モデル	$M_E$	$H(t) = N(1 - \exp(-\phi t))$	$N$ $\phi$
	$M_{HE}$	$H(t) = \sum_{i=1}^n N_i(1 - \exp(-\phi_i t))$	$N_i$ $\phi_i$
S字型モデル	$M_{DS}$	$H(t) = N(1 - (1 + \phi t)\exp(-\phi t))$	$N$ $\phi$
	$M_{IS}$	$H(t) = N \frac{1 - \exp(-\phi t)}{1 + \frac{r}{r} \exp(-\phi t)}$	$N$ $\phi$ $r (> 0)$

ponential)モデル $M_E$ 、大場ら<sup>[6, 13]</sup>が提案した HE(Hyperexponential)モデル $M_{HE}$ などがある。一方、S字型SRGMには、山田ら<sup>[11]</sup>が提案した DS(Delayed S-shaped)モデル $M_{DS}$ 、大場<sup>[3, 6, 8]</sup>が提案した IS(Inflection S-shaped)モデル $M_{IS}$ などがある。

各モデルにおける信頼度成長曲線  $H(t)$  の定義式を表1にまとめる。表1中で、 $t$ はテスト中の時刻、 $N$ はテスト開始時にプログラム中に存在したフォールトの総数 (初期フォールト数)、 $\phi$ はフォールトの発見率、 $r$ はフォールトの総数に対する発見可能なフォールトの総数の割合を表す。なお、HEモデル $M_{HE}$ は幾つかのクラスタ (機能的にまとまったモジュール群) から構成されるプログラムの解析に用いられる。そこで、 $N_i$ 、 $\phi_i$ は、それぞれ、クラスタ  $i$  に関する  $N$ 、 $\phi$  である。

本稿では、5人の学生によるコンパイラ作成の学生実験から収集したデータを用いて、上述の4つのSRGM,  $M_E$ ,  $M_{HE}$ ,  $M_{DS}$ ,  $M_{IS}$ の比較を行う。

### 3. 実験データ

情報工学専攻の約40人の学生を対象に実験データを収集した。ここでは、予め定めた期間内に実験を終了した5人の学生(x125, x132, x133, x134, x136)に関する実験データを利用する。

各学生は言語Pascalの部分集合に対して、言語Cを用いてコンパイラを作成した。どの学生もコンパイラ作成に関する講義を聴講しているが、実際の作成経験はなかった。作成されたプログラムの大きさは行数にして約1000行であった。プログラム作成とプログラムテストは同一の学生が行った。なお、最終的なフォールト数を求めるため、実験終了後に筆者らがプログラムテストを更に行った。

表2に5人の学生から収集した実験データを示す。 $(\tilde{t}, \tilde{N})$ の $\tilde{t}$ は観測時刻、 $\tilde{N}$ は時刻 $\tilde{t}$ までに発見されたフォールトの累積数を表す。 $N_r$ は実験終了後にプログラム中に残っていたフォールトの総数である。なお、本実験では $\tilde{t}$ は累積端末使用時間で表す。 $\tilde{N}$ の値はプログラムテキストの更新記録を分析して求めた。 $N_r$ の値は学生実験終了後に筆者らが行ったプログラムテストによって求めた。

表2の実験データの主な特徴を次に示す。

- C1 5組の実験データが利用できる。
- C2 フォールト発見時刻が分単位で求められている。これは、ソフトウェアの信頼度評価の通常の実験データに比べて詳細である。
- C3 プログラムテストの期間が短い。(高々、3803分=2.5日の長さである。)
- C4 プログラムの大きさが比較的小さい。(約1000行)
- C5 プログラムが2種類のクラスタ(構文解析部とコード生成部)から構成されている。この構成に対応させて表2を分けると、表3が求まる。

### 4. 指数型モデルとS字型モデルとの比較

ここでは、指数型モデル $M_E$ とS字型モデル $M_{DS}$ ,  $M_{IS}$ の間の比較を行う。

#### 4.1 パラメータの推定

表1に示す様に、関数 $H(t)$ を定めるにはパラメータ $N$ ,  $\phi$ , ( $M_{IS}$ に対しては) $r$ を求める必要がある。なお、 $N$ と $\phi$ は実験データに基づいて最ゆう推定法によって求めた(詳細は文献[6]の付録A-Cを参照されたい)。

パラメータ $r$ はフォールトの総数に対する発見可能なフォールトの総数の割合である<sup>[4]</sup>。 $r$ の値の決定

は、一般的に、非常に困難である。実際には、過去の経験やデータに基づいて $r$ を決めるべきである。今回の実験では、作成するプログラムが構文解析部とコード生成部の2つの部分から構成されている事実に注目した。しかも、構文解析部の後にコード生成部が位置している。そこで、構文解析部のフォールトの総数が、近似的に、発見可能なフォールトの総数に等しいと仮定した。よって、フォールトの総数で構文解析部のフォールトの総数を割った値を $r$ と定める。

パラメータ $N$ ,  $\phi$ ,  $r$ の計算を次の2つの場合について行った。表2の全ての実験データを利用した計算を場合1、表2の5組のデータのそれぞれに対し、最初の15個のデータだけを利用した計算を場合2と呼ぶ。

#### 4.2 モデルの適用

比較の基準として次の2つを考える。

- (1) 初期フォールト数に対する推定の精度  $A$

$$A = |N_0 - N| / N_0$$

ここで、 $N_0$ ,  $N$ はそれぞれ初期フォールト数の実測値、推定値とする。

- (2) カイ2乗適合度検定による信頼度成長曲線 $H(t)$ の適合度  $\chi^2$

モデル $M_E$ ,  $M_{DS}$ ,  $M_{IS}$ を5人の学生から収集した実験データに適用した結果を、それぞれ、表4、表5、表6に示す。 $A$ は、5人の学生に対する $A$ の平均を示す。 $p$ はカイ2乗適合度検定で $H(t)$ が実測値と適合していると判断された割合を示す。

#### 4.3 比較結果

- (1) 初期フォールト数の推定精度

フォールトに関する全ての実験データを利用する場合(場合1)、表4(a)、表5(a)、表6(a)中の $\bar{A}$ を見ると、 $M_E$ と $M_{IS}$ が $M_{DS}$ より優れていることが分かる。しかも、 $M_E$ と $M_{IS}$ における $\bar{A}$ の値は5%以下で、精度が非常に高い。

最初の15個のデータを用いた場合(場合2)では $M_E$ ,  $M_{IS}$ が共に27%程度に大きくなったのに対し、 $M_{DS}$ は場合1とほぼ同じ値であった。3つのモデルに共通しているのは、場合1に比べて推定値が小さくなることである。

- (2) 信頼度成長曲線 $H(t)$ の適合度

S字型モデルの $M_{DS}$ が $p=5/5$ ,  $M_{IS}$ が $p=4/5$ である。これに対し指数型モデル $M_E$ は $p=1/5$ である。従って、S字型モデルが指数型モデルより優れていることが分かる。場合2に関しては、どのモデルも場合1と同じ結果となった。

- (3) 総合評価

(1), (2)で述べた結果から見ると、S字型モデルが指数型モデルより優れていると考えられる。但し、S字型モデル $M_{DS}$ と $M_{IS}$ のどちらがより優れているかを決めることは困難である。学生x132, x134に対する各モデルの信頼度成長曲線 $H(t)$ を図2、図3に示す。

表2. 実験データ

X125 ( $\bar{i}, \bar{N}$ )	X132 ( $\bar{i}, \bar{N}$ )	X133 ( $\bar{i}, \bar{N}$ )	X134 ( $\bar{i}, \bar{N}$ )	X136 ( $\bar{i}, \bar{N}$ )
3536 1	1245 1	2159 1	3334 1	3403 1
3540 2	1289 2	2173 2	3471 2	3476 2
3545 3	1456 3	2201 3	3488 3	3599 3
3552 4	1537 5	2254 4	3495 4	3705 4
3554 5	1631 6	2257 5	3567 5	3734 6
3602 6	1721 7	2273 6	3621 7	3750 7
3608 8	1734 8	2299 7	3674 8	3783 8
3672 9	1742 9	2310 8	3709 10	3791 9
3697 10	1967 10	2452 9	3728 11	3891 10
3700 11	2009 11	2518 10	3765 12	4194 11
3742 12	2013 12	2531 11	3897 13	4273 13
3826 13	2069 13	2681 12	3902 14	4309 14
3991 14	2075 14	2697 13	3962 15	4370 15
4014 19	2254 15	2730 14	4174 16	4396 16
4037 20	2316 16	2741 15	4274 17	4457 17
4080 21	2337 17	2746 16	4428 18	4458 18
4177 22	2554 18	2748 17	4449 19	4646 20
4178 24	2627 19	2779 18	4509 20	4786 21
4232 25	2638 21	2896 19	4524 21	4825 22
4279 26	2673 22	2901 20	4535 22	5260 23
4294 27	3059 23	2966 21	5395 23	5267 24
	3341 24		5419 24	5365 25
			6219 25	5433 26
			6903 27	5830 27
$N_r=5$	$N_r=1$	$N_r=2$	$N_r=1$	$N_r=1$

表3. 構文解析部, コード生成部に対する( $\bar{i}, \bar{N}$ )

	X125 ( $\bar{i}, \bar{N}$ )	X132 ( $\bar{i}, \bar{N}$ )	X133 ( $\bar{i}, \bar{N}$ )	X134 ( $\bar{i}, \bar{N}$ )	X136 ( $\bar{i}, \bar{N}$ )
構文解析部	3540 1	1245 1	2159 1	3334 1	3403 1
	3552 2	1289 2	2173 2	3488 2	3599 2
	3554 3	1456 3	2201 3	4274 3	3791 3
	3602 4	1631 4	2254 4	4428 4	3891 4
	3608 6	1721 5	2257 5	4449 5	4194 5
	3672 7	1742 6	2299 6	6903 7	4273 6
	3697 8	2254 7	2452 7		4309 7
	3700 9	2316 8	2531 8		4370 8
	3742 10	2337 9	2681 9	3471 1	4786 9
	3826 11	2627 10	2730 10	3495 2	5260 10
	3991 12	2638 11	2741 11	3567 3	5267 11
	4014 16	2673 12	2746 12	3621 5	5365 12
	4037 17	3341 13	2966 13	3674 6	5433 13
	4080 18			3709 8	5830 14
	4177 19			3728 9	
	4178 21			3765 10	3476 1
	4232 22			3897 11	3705 2
4279 23			3902 12	3734 4	
4294 24			3962 13	3750 5	
コード生成部			4174 14	3783 6	
			4509 15	4273 7	
			4524 16	4396 8	
			4535 17	4457 9	
	3536 1	1537 2		4458 10	
	3545 2	1734 3	2273 1	4646 12	
	4014 3	1967 4	2310 2	4825 13	
		2009 5	2518 3		
		2013 6	2518 3		
		2069 7	2697 4		
		2075 8	2748 5		
	2554 9	2779 6			
	2638 10	2896 7			
	3059 11	2901 8			

表4. モデル  $M_E$  の適用結果

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	30.4	26.6	24.9	28.0	29.4	
$\phi$	0.00274	0.00104	0.00191	0.00088	0.00099	
$A$	5.0%	6.4%	8.3%	0.0%	5.0%	$\bar{A}=4.94\%$
$\chi^2$	○	×	×	×	×	$p=1/5$

(a) 場合 1

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	22.7	19.0	17.8	19.8	20.1	
$\phi$	0.00399	0.00152	0.00249	0.00167	0.00160	
$A$	29.1%	24.0%	22.6%	29.1%	28.2%	$\bar{A}=26.60\%$
$\chi^2$	○	×	×	×	×	$p=1/5$

(b) 場合 2

表5. モデル  $M_{DS}$  の適用結果

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	31.6	26.7	33.2	27.3	29.0	
$\phi$	0.00429	0.00174	0.00222	0.00172	0.00174	
$A$	1.3%	6.8%	44.3%	2.5%	3.6%	$\bar{A}=11.70\%$
$\chi^2$	○	○	○	○	○	$p=5/5$

(a) 場合 1

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	23.8	25.0	23.5	23.2	26.1	
$\phi$	0.00612	0.00179	0.00293	0.00221	0.00192	
$A$	25.6%	0.0%	2.2%	17.1%	6.8%	$\bar{A}=10.34\%$
$\chi^2$	○	○	○	○	○	$p=5/5$

(b) 場合 2

表6. モデル  $M_{Te}$  の適用結果

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	34.6	25.8	24.7	27.2	28.6	
$\phi$	0.00203	0.00144	0.00239	0.00165	0.00139	
$r$	0.89	0.54	0.62	0.26	0.52	
$A$	8.1%	3.2%	7.4%	2.9%	2.1%	$\bar{A}=4.74\%$
$\chi^2$	○	○	×	○	○	$p=4/5$

(a) 場合 1

	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N$	24.1	18.2	18.5	17.9	19.6	
$\phi$	0.00367	0.00232	0.00260	0.00426	0.00225	
$r$	0.80	0.47	0.73	0.13	0.53	
$A$	24.7%	27.2%	19.6%	36.1%	30.0%	$\bar{A}=27.52\%$
$\chi^2$	○	○	×	○	○	$p=4/5$

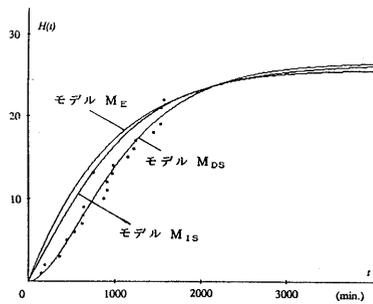
(b) 場合 2

表7. モデル  $M_{HE}$  の適用結果

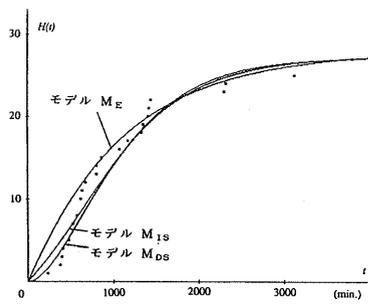
	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N_p$	27.5	14.3	14.7	7.29	15.9	
$\phi_p$	0.00259	0.00109	0.00224	0.00085	0.00084	
$N_G$	3.0	12.5	10.4	20.8	15.2	
$\phi_G$	0.01980	0.00109	0.00163	0.00109	0.00126	
$A$	4.6%	6.9%	9.1%	0.4%	11.1%	$\bar{A}=6.42\%$
$\chi^2$	○	×	×	×	×	$p=1/5$

表8. 新しい方法によるモデル  $M_{HE}$  の適用結果

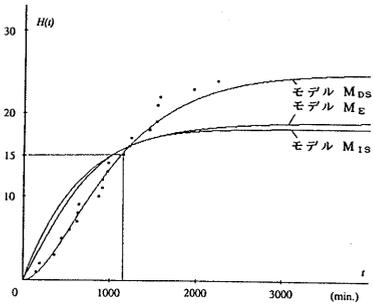
	x125	x132	x133	x134	x136	
$N_o$	32	25	23	28	28	
$N_p$	27.5	14.3	14.7	7.29	15.9	
$\phi_p$	0.00259	0.00109	0.00224	0.00085	0.00084	
$N_G$	3.0	11.6	9.8	20.3	15.1	
$\phi_G$	0.01980	0.00186	0.00242	0.00151	0.00134	
$A$	4.6%	3.6%	10.8%	1.5%	10.6%	$\bar{A}=6.22\%$
$\chi^2$	○	○	×	○	×	$p=3/5$



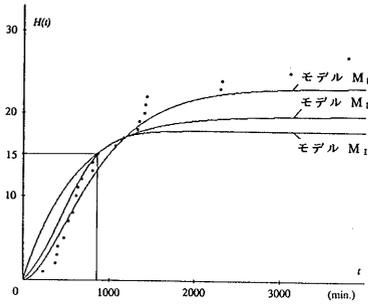
(a) 場合 1



(a) 場合 1



(b) 場合 2



(b) 場合 2

図2. 信頼度成長曲線の比較 (x132)

図3. 信頼度成長曲線の比較 (x134)

## 5. HEモデルに対する新しい提案

ここでは、モデル $M_{HE}$ に対する新しい適用方法を提案する。この方法は信頼度成長曲線 $H(t)$ の原点の取り方に特徴がある。

### 5.1 $M_{HE}$ と $M_E$ の比較

$M_{HE}$ は幾つかのクラスタから構成されるプログラムに適用される。各クラスタ毎に、指数型信頼度成長曲線を求め、それを単純に加え合わせてプログラム全体を評価する。その理由は、クラスタ毎に初期フォールト数、フォールトの発見率が異なることを考慮するためである。例えば、プログラムが新しく作成したクラスタと再利用したクラスタから構成されるような場合に $M_{HE}$ モデルは有効である<sup>[9]</sup>。文献[6,13]によると、 $M_{HE}$ における $H(t)$ は次のように定義される。

$$H(t) = \sum_{i=1}^n N_i \{1 - \exp(-\phi_i t)\}$$

$n$  : クラスタの数

$N_i$  : クラスタ  $i$  の初期フォールト数

$\phi_i$  : クラスタ  $i$  のフォールト発見率

理論的には、クラスタ数が1ならば $M_{HE}$ と $M_E$ は等価である。クラスタ数が多くなればなるほど、信頼度成長曲線の形状は指数型(Exponential)からはなれ、超指数型(Hyperexponential)となる。

表7に $M_{HE}$ の適用結果を示す。表7中の $N_p$ 、 $\phi_p$ は構文解析部の初期フォールト数、フォールト発見率を表す。 $N_G$ 、 $\phi_G$ はコード生成部の初期フォールト数、フォールト発見率を表す。表7と表4(a)を比較すると、このデータに関する限り、 $M_E$ が $M_{HE}$ より優れていることが分る。1つの理由として、フォールトをクラスタ毎に分けたため、フォールト数がパラメータを推定するのに十分なだけの大きさをなくなってしまったことが考えられる。

しかし、筆者らはプログラムテストの開始時刻がクラスタ毎に異なることが考慮されていないことに注目する。これが $M_{HE}$ の精度を下げる主要原因であったとする立場をとる。

注意1: この実験の場合、クラスタ毎にテスト開始時刻を測ることが可能であった。

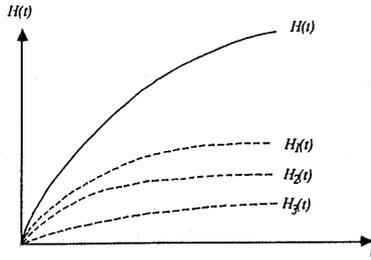


図4.  $H_i(t)$ の原点が同一と仮定した場合のモデル  $M_{HE}$  の信頼度成長曲線

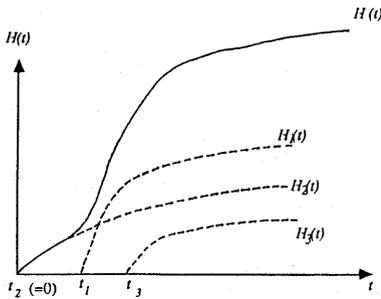
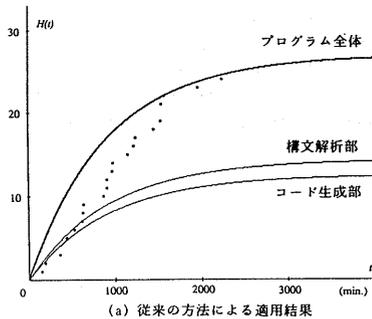
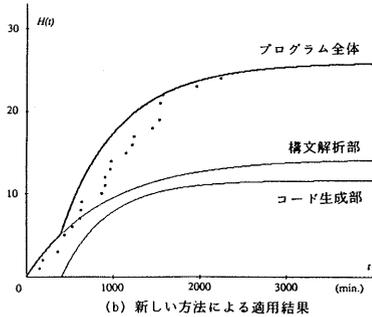


図5.  $H_i(t)$ の原点が異なると仮定した場合のモデル  $M_{HE}$  の信頼度成長曲線



(a) 従来の方法による適用結果



(b) 新しい方法による適用結果

図6. モデル  $M_{HE}$  の信頼度成長曲線 (x132)

## 5.2 新しい適用法

図4に示す様に、 $M_{HE}$ ではクラスタ毎に $N_i$ と $\phi_i$ を求め、求めた信頼度成長曲線 $H_i(t)$ を単純に加え合わせる。このとき、各信頼度成長曲線の原点は同一であると仮定している。

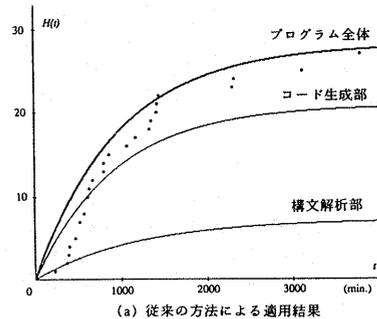
これに対し、提案する方法では図5に示す様に、クラスタ毎に信頼度成長曲線 $H_i(t)$ の原点が異なるものとする。こうして $H_i(t)$ を加え合わせると、信頼度成長曲線 $H(t)$ はS字型モデルの曲線に似てくる。

**注意2:** 各クラスタ毎の信頼度成長曲線の原点のズレを考慮することの重要性について、同様の指摘が文献[13]にある。但し、上述の手順の記述、及び、5.3で述べる様に具体的なデータを用いた解析例等の報告は、筆者らの知る限り、ない。

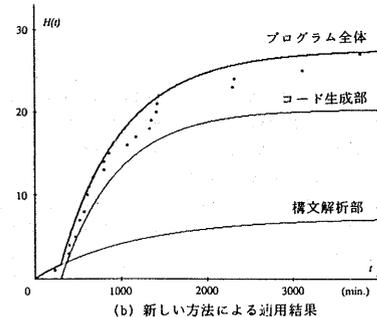
## 5.3 適用例

5.2で説明した考えに基づき、 $M_{HE}$ を適用し直してみる。今回の実験で対象としたプログラムは2つのクラスタ(構文解析部とコード生成部)から構成されている。更に、構文解析部が(フォールトを含むことを許したとしても)少なくとも動作するまでコード生成部のテストはできない。従って、構文解析部が動作し始めた時刻をコード生成部のテスト開始時刻とみなす。一方、構文解析部のテスト開始時刻については、プログラムテキストの更新記録と各学生へのインタビューを通じて決定した。

表8に $M_{HE}$ の新しい適用方法による結果を示す。表7と比べると、Aの値はほぼ同じであるが、pの値が改善されていることが分かる。学生x132, x134に関して、適用方法の違いによる信頼度成長曲線 $H(t)$ の差を図6、図7に示す。



(a) 従来の方法による適用結果



(b) 新しい方法による適用結果

図7. モデル  $M_{HE}$  の信頼度成長曲線 (x134)

## 6. あとがき

本稿ではソフトウェア信頼度成長モデルの実験的評価を行った。実験データは言語Pascalの部分集合に対してコンパイラを作成する学生実験から収集したものである。一般的なモデル評価実験に比べると、プログラムテストの期間が短く、作成されるプログラムの大きさも小さい。また、フォールトの発見時刻も分単位で、詳細なものである。

評価の結果、初期フォールト数の推定の誤差、及び、信頼度成長曲線の適合度に関しS字型モデルが指数型モデルより優れていることが分った。

次に、実用的立場から、テストの初期段階のデータのみを用いてモデルを適用する試みを行った。その結果、すべてのモデルにおいて、初期フォールト数の推定値が、全データを用いた場合よりも小さくなることが分かった。

最後に、超指数型信頼度成長モデル $M_{HE}$ に対し、推定の精度を高める新しい方法を提案した。この方法に従って $M_{HE}$ を適用し直した結果、信頼度成長曲線の適合度が改善された。また、信頼度成長曲線の形状もS字型モデルに近くなった。テスト開始時刻がプログラムのクラスタ毎に異なるとする考え方は、プログラム中のフォールトが相互に依存しているとするS字型モデル $M_{IS}$ の立場と密接な関係があると考えられる。

$M_{HE}$ に対する新しい方法は、現在のところ、小規模なプロジェクトに限定して有効である。大規模なプロジェクトであればテスト期間そのものが長いので、信頼度成長曲線の原点の位置はあまり本質的でない。また、クラスタ毎にテスト開始時刻を測る必要があるが、一般にそれが可能であるとは限らない。

現在、新年度の学生実験を対象として、ソフトウェア信頼度成長モデルの実験的評価を行っている。実験では、プログラムの設計、コーディング段階のフォールトデータに対するモデルの適用、HEモデルに対して提案した方法の他のモデルへの応用などについても検討している。

謝辞：本評価において快く被験者を引き受けていただいた大阪大学基礎工学部情報工学科の方々に感謝致します。更に、本報告の作成において、原稿の清書にご協力いただいた本学学部生 楠本 真二君、児玉 寛君に感謝致します。

## 参考文献

- [1] A.L.Goel, "Software reliability models : Assumptions, limitations, and applicability," IEEE Trans. Software Eng., vol. SE-11, pp.1411-1423, Dec. 1985.
- [2] A.L.Goel and K.Okumoto, "Time-dependent error detection rate model for software reliability and other performance

measures," IEEE Trans. Rel., vol. R-28, pp.206-211, Aug. 1979.

- [3] 梶山 他, "テストの習熟を考慮したソフトウェア信頼度成長モデル," 情報処理学会第25回全国大会, 1E-3, pp.401-402, 1982.
- [4] K.Kishida et al., "Quality - assurance technology in Japan," IEEE Software, vol.4, no.5, pp.11-18, Sep. 1987.
- [5] K.Matsumoto et al., "Experimental evaluation of software reliability growth models," (Submitted to FTCS-18).
- [6] M.Ohba, "Software reliability analysis models," IBM J. Res. Develop., vol.28, no.4, pp.428-443, July 1984.
- [7] 大場 他, "ソフトウェア・エラー推定法の比較," 情報処理学会第25回全国大会, 1E-4, pp.403-404, 1982.
- [8] 大場 他, "習熟型ソフトウェア信頼度成長モデル," 情処研報, SW-28-6, 1983.
- [9] Y.Tohma et al., "Structural approach to the estimation of the number of residual software faults based on the hypergeometric distribution," Proc. COMPSAC87, pp.361-367, Oct. 1987.
- [10] 山本浩司, "ソフトウェアの残存エラー数推定モデルの定量的比較," 大阪大学基礎工学部情報工学科特別研究報告, 1987.
- [11] S.Yamada, M.Ohba and S.Osaki, "S-shaped reliability growth modeling for software error detection," IEEE Trans. Rel., vol. R-32, pp.475-478, Dec. 1983.
- [12] S.Yamada, and S.Osaki, "Software reliability growth modeling : Models and applications," IEEE Trans. Software Eng., vol. SE-11, pp. 1431- 1437, Dec. 1985.
- [13] 米原 他, "モジュール構造ソフトウェアの信頼度成長曲線," 情報処理学会第25回全国大会, 1E-2, pp.399-400, 1982.