

右側打切りデータを用いた ソフトウェアの適応的若化方策

林 坂 弘 一 郎*¹

ウォン エリック*²

土 肥 正*³

概要

本論文ではエージング現象に起因するシステム障害が発生するようなソフトウェアシステムに対して、その障害時間データと右側打切り時間データが得られる状況を想定し、最適若化スケジュールをノンパラメトリック予測推論によって導出する。具体的には u 個の障害時間データおよび v 個の右側打切り時間データから構成される $n = u + v$ 個のランダム打切りデータから、 $n + 1$ 番目の障害発生時間の確率分布と予測信頼度関数の上下限をノンパラメトリック予測推論によって定式化し、システムアベイラビリティを最大にするソフトウェア若化スケジュールを適応的に導出する。また、シミュレーション実験を通じて、新たなデータを逐次得ることで最適若化スケジュールの推定値が更新され、理論的な最適解へ収束することを示す。

キーワード：ソフトウェアエージング、ソフトウェア若化、ランダム打切り、ノンパラメトリック予測推論、アベイラビリティ

1. まえがき

ソフトウェアシステムが長期間にわたって連続して動作するとき、時間の経過とともにエラーが累積することで劣化が進み、システム障害へと至ることがある。ソフトウェアエージング (software aging) と呼ばれるこのような劣化現象は、オペレーティングシステム、ミドルウェアシステム、サーバソフトウェアなどの数多くのシステムで観測されており、ソフトウェ

*¹ 本学経営学部准教授

*² テキサス大学ダラス校コンピュータサイエンス学科教授

*³ 広島大学大学院工学研究院教授

アエージングに起因する障害も報告されている [1-14].

ソフトウェアエージングの原因となるフォールトはエージング関連バグ (aging-related bugs) と呼ばれている. 例えば, メモリリーク, ファイルロック解除漏れ, ストレージのフラグメンテーション, 内部変数の丸め誤差, スレッドの暴走などが累積することで, 徐々に劣化が進み, パフォーマンスの劣化やシステムのクラッシュといったシステム障害へ至ることになる. エージング関連バグは, この根本原因をソフトウェアシステムのソースコード上で特定することが極めて困難であることから, その対処法について数多くの研究がなされてきた. 特に, ソフトウェア若化 (software rejuvenation) と呼ばれる方策はエージング関連バグに起因する障害を予防するための有効な方法として認識されている [15]. ソフトウェア若化はソフトウェアシステムの稼働を一時的に停止し, その内部構造を浄化したあとにシステムを再稼働するという一連の予防保全手続きを意味する. ここで, ソフトウェアシステムの内部構造の浄化とは, ガーベジコレクションやオペレーティングシステムにおけるカーネルテーブルの洗浄, データ構造の初期化などを意味し, 最も頻繁に行われているソフトウェア若化の一例がハードウェアリブートである.

ここで問題となるのは, どのようなタイミングでシステムを予防的に若化するかにある. Huang ら [15] はソフトウェアシステムの時間的挙動を, 正常稼働状態, 障害発生可能な状態, 障害の発生状態, ソフトウェア若化状態の 4 状態をもつ連続時間マルコフ連鎖で記述した. Dohi ら [16,17] は Huang ら [15] の連続時間マルコフ連鎖モデルを連続時間および離散時間のセミマルコフモデルに拡張し, 更に障害発生時間データから経験分布関数に基づいた標準総試験時間統計量を定義することにより最適な若化スケジュールを推定する統計的手法を開発している. Vaidyanathan ら [18] はシステム障害に至るまでの時間分布に 2 次アーラン分布を考えたセミマルコフモデルに対して, アベイラビリティとダウンタイムコストの観点から最適ソフトウェア若化スケジュールの導出を議論した. また, Rinsaka and Dohi [19] はセミマルコフモデルで記述されたモデルに対して, Rinsaka and Dohi [20] はマルコフ更新過程 (Markov Regenerative Process) で記述されたモデルに対して最適ソフトウェア若化スケジュールを核密度推定によって少数の障害発生時間データから高精度に推定するアルゴリズムを提案した. Zhao ら [21] はメモリリークの注入実験による加速寿命試験を行い, セミマルコフモデルに基づいた最適若化スケジュールを導出するためのパラメトリック/ノンパラメトリックモデルを提案した.

一方でノンパラメトリック予測推論 (nonparametric predictive inference) [22] は Hill の仮定 $A_{(n)}$ [23,24] に基づき, n 個のデータが観測された後の $n+1$ 番目の事象に対して予測を行う統計手法である. Coolen and Newby [25], Coolen and Coolen-Schrijner [26], Coolen-

Schrijner and Coolen [27] はハードウェアの予防取替問題に対してノンパラメトリック予測推論を適用した。Coolen-Schrijner and Coolen [28] は $n+1$ 番目及び $n+2$ 番目の故障時間を予測し最適予防取替年齢を適応的に求めている。Coolen and Coolen-Maturi [29] は冗長化システムにおける複数の部品が同時に故障するような状況にノンパラメトリック予測推論を適用している。さらに、Coolen and Yan [30] は打ち切りを含んだ故障データに対するノンパラメトリック予測推論を提案し、Kaplan–Meier 法 [31] や Berliner–Hill 法 [32] との比較を行った。

林坂, 土肥 [33] は文献 [19] と同様のセミマルコフモデルに対してノンパラメトリック予測推論を適用し, $n+1$ 番目及び $n+2$ 番目の障害時間を予測するとともに, 適応的にシステムアベイラビリティを最大化するソフトウェア若化スケジューリングを実施した。また, 林坂, 土肥 [34] は $n+3$ 番目の障害時間を取り扱うように拡張した。さらに, Rinsaka and Dohi [35–37] は文献 [20] と同様のマルコフ更新過程によるモデルに対してもノンパラメトリック予測推論を適用した。文献 [35] では $n+1$ 番目の障害時間を予測し, 文献 [36,37] では $n+1$ 番目, $n+2$ 番目の障害時間を予測し, システムアベイラビリティを最大化する若化スケジュールを導出した。

本論文では文献 [33,34] で提案された 2 ステップおよび 3 ステップの適応的若化スケジューリングを一般的な適応的若化スケジューリングへと拡張する。すなわち, u 個の障害発生時間データと v 個の右側打ち切り時間データから構成される $n = u + v$ 個のランダム打ち切りデータからノンパラメトリック予測推論によって $n+1$ 番目の障害時間を予測し, システムアベイラビリティの上下限を最大化する若化スケジュールを導出する。これにより, オンラインでシステム障害時間と右側打ち切り時間データを逐次取得しながら, 次のシステム障害に対して適応的に最適若化スケジュールを導出できるようになる。また, 2 種類のシミュレーション実験によって本論文で提案する適応的ソフトウェア若化スケジュールの特徴について考察し, その有効性を検証する。

2. セミマルコフモデル

本論文では, ソフトウェアシステムが 4 状態をもつセミマルコフモデルについて考える。ソフトウェアシステムは状態 0 の正常稼動状態からスタートし, メモリリーク量があるしきい値を超えるなどして状態 1 の劣化状態に移移する。ここで状態 1 は正常稼動状態と比べて不安定でシステムが障害発生可能な状態である。状態 0 から状態 1 に移移する時間 Z の確率分布関数を $\Pr\{Z \leq t\} = F_0(t)$, 平均を $\mu_0 (> 0)$ とする。システムが状態 1 から状態 2 の障害発生状態に移移するまでの時間を非負で連続な確率変数 X とし, その確率分布関数を

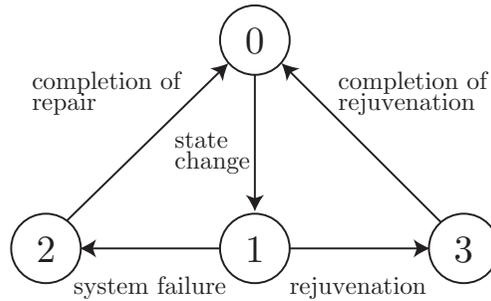


図1 モデルの状態推移図

$\Pr\{X \leq t\} = F_f(t)$, 信頼度関数を $S(t) = 1 - F_f(t)$, 平均を $\lambda_f (> 0)$ とする. 一旦障害が発生すると, 事後保全が直ちに開始される. 事後保全とは, 障害が発生した後に事後的にソフトウェアシステムを若化することを意味しており, 本論文では修復という言葉によって代用する. 修復に要する時間 Y も非負で連続な確率変数であり, その確率分布関数を $\Pr\{Y \leq t\} = F_a(t)$, 平均を $\mu_a (> 0)$ とする. 修復完了後は正常稼働状態である状態 0 へ戻る.

一方, システムが状態 1 の劣化状態に推移すると予防的なソフトウェア若化を行うかどうかの決定をするものとする. 状態 1 から予防的な若化によって状態 3 の予防保全状態に推移するまでの時間 T もまた非負で連続な確率変数であり, その確率分布関数を $F_r(t)$, 平均を $t_0 (> 0)$ とする. 予防的な若化にもオーバーヘッドが存在し, そのオーバーヘッド V に対する確率関数を $\Pr\{V \leq t\} = F_c(t)$, 平均を $\mu_c (0 < \mu_c < \mu_a)$ とする. 修復の場合と同様に, 予防的な若化が完了すると, システムの障害発生率は正常稼働状態における初期状態まで復旧される. このモデルは状態 0 から再度状態 0 に戻るまでの時間間隔を周期にもつセミマルコフ過程であり, すべての状態が再生点 (regeneration points) となる. システムの状態推移図を図 1 に示す. 特に状態 1 から予防的に若化を実施するまでの時間が t_0 の一定であるとすれば, 確率分布関数 $F_r(t)$ を次のようなユニット関数

$$F_r(t) = U(t - t_0) = \begin{cases} 1 & \text{if } t \geq t_0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

に置き換えればよい.

Dohi ら [16] の結果を直接用いることによって, 定常アベイラビリティは

$$\begin{aligned} A(t_0) &= \Pr\{\text{software system is operative in the steady state}\} \\ &= \frac{\mu_0 + \int_0^{t_0} S(t)dt}{\mu_0 + \mu_a[1 - S(t_0)] + \mu_c S(t_0) + \int_0^{t_0} S(t)dt} \end{aligned} \quad (2)$$

のように求めることができる。以下では、式 (2) の $S(\cdot)$ に対して、理論分布を仮定することなくシステム障害発生時間データと打ち切り時間データからノンパラメトリック予測推論に基づいた予測を行い、式 (2) のシステムアベイラビリティを最大にする最適ソフトウェア若化スケジュールを適応的に導出することを考える。

3. 右側打ち切りデータを用いたノンパラメトリック予測推論

ここでは右側打ち切りを考慮したノンパラメトリック推論による信頼度関数の上下限について議論する。ノンパラメトリック予測推論は Coolen ら [22] が Hill [23, 24] の仮定 $A_{(n)}$ から導出した統計手法である。今、 $X_i, i = 1, 2, \dots, n$ を交換可能な連続型の確率変数とし、 $x_{(i)}$ をその順序統計量とすると、Hill の仮定 $A_{(n)}$ は次の通りである。

- (A-1) タイは起こらない。つまり $i \neq j$ に対して、 $x_{(i)} \neq x_{(j)}$ である。
 (A-2) $x_{(i)}, i = 1, 2, \dots, n$ が与えられるとき、次の実現値が区間 $(x_{(i)}, x_{(i+1)})$, $i = 0, 1, \dots, n$ となる確率は $1/(n+1)$ である。ただし、 $x_{(0)} = 0$ かつ $x_{(n+1)} = \infty$ (あるいは X_{n+1} の上限が存在する) とする。

つまり、Hill の仮定 $A_{(n)}$ は連続した障害時刻間に次の障害が発生する予測確率を与えている。

Coolen and Yan [30] によって与えられた $A_{(n)}$ に右側打ち切りを考慮した一般化 rc- $A_{(n)}$ は、区間に入れ子構造を認めた M 関数を導入することで、次の実現値の予測確率を記述している。すなわち、Coolen and Yan [30] は T の次の実現値がある区間 (a, b) となる部分的な確率を区間 (a, b) に対する M 関数、 $M_T(a, b)$ と定義した。 M 関数から得られる M 関数値一つひとつは $[0, 1]$ の値をとり、 T に対する定義域すべての区間における M 関数値の合計は必ず 1 になる。ここで、区間 (a, b) の入れ子構造は右側打ち切りを含むデータに対して適用する際に利用される。打ち切りデータが含まれない n 個の完全データに対しては、仮定 $A_{(n)}$ に基づく X_{n+1} の予測確率は次式で与えられる。

$$M_{X_{n+1}}(x_{(i)}, x_{(i+1)}) = \frac{1}{n+1}, \quad i = 0, 1, \dots, n. \quad (3)$$

次に右側打ち切りを含むデータに対する予測分布を考える。このために以下のような記号を定義する。まず、 n 個の交換可能な非負の連続型確率量 X_1, X_2, \dots, X_n を考える。そのうち u ($0 \leq u \leq n$) 個については真の障害時間が観測されており、その順序統計量を $0 < t_{(1)} < t_{(2)} < \dots < t_{(u)}$ とする。一方で、 $v = n - u$ 個については真の障害時間が観測されることなく、予防若化の実施などにより、右側打ち切り時間のみが観測され、その順序統計量を

$0 < c_{(1)} < c_{(2)} < \cdots < c_{(v)}$ とする. さらに, $i = 1, 2, \dots, n$ に対して, $(x_{(i)}, \delta_{(i)})$ の組を定義する. ここで, $x_{(1)} < x_{(2)} < \cdots < x_{(n)}$ は u 個の障害時間と v 個の右側打ち切り時間から構成される順序統計量であり,

$$\delta_{(i)} = \begin{cases} 1 & x_{(i)} \text{ が障害時間} \\ 0 & x_{(i)} \text{ が右側打ち切り時間} \end{cases} \quad (4)$$

とする. なお, $x_{(0)} = 0, \delta_{(0)} = 1, x_{(n+1)} = \infty$ (あるいは上限が存在する), $\delta_{(n+1)} = 1$ とする. また, 障害時間と打ち切り時間の独立性を仮定するとともに, 仮定 $A_{(n)}$ と同様に障害時間と打ち切り時間にもタイは存在しない, すなわち, $i \neq j$ に対して $x_{(j)} \neq x_{(i)}$ を仮定する.

今, $i = 0, 1, \dots, u$ に対して, 時間区間 $I_i = (t_{(i)}, t_{(i+1)})$ を定義し, 時間区間 I_i に発生した右側打ち切り時間の順序統計量を $c_1^i < c_2^i < \cdots < c_{l_i}^i$ とする. なお, $l_i (= 0, 1, \dots)$ は時間区間 I_i での右側打ち切り回数を意味する. したがって, $\sum_i l_i = v$ である. また \tilde{n}_t は時刻 t 直前までソフトウェアシステムが動作した回数を表すものとする. つまり, 時刻 t で障害や打ち切りが観測されたとき, \tilde{n}_t には, その障害または打ち切りの回数も含まれることになる.

これまでの定義から, u 個の障害時間と v 個の打ち切り時間データが得られるとき, Coolen and Yan [30] によって提案された仮定 rc- $A_{(n)}$ に基づき, X_{n+1} の実現値が時間区間 $(t_{(i)}, t_{(i+1)})$ または $(c_k^i, t_{(i+1)})$ となる部分的な確率は次のような M 関数によって与えられる.

$$M_{X_{n+1}}(t_{(i)}, t_{(i+1)}) = \frac{1}{n+1} \prod_{\{r:c_{(r)} < t_{(i)}\}} \frac{\tilde{n}_{c_{(r)}} + 1}{\tilde{n}_{c_{(r)}}}, \quad (5)$$

$$M_{X_{n+1}}(c_k^i, t_{(i+1)}) = \frac{1}{(n+1)\tilde{n}_{c_k^i}} \prod_{\{r:c_{(r)} < c_k^i\}} \frac{\tilde{n}_{c_{(r)}} + 1}{\tilde{n}_{c_{(r)}}}. \quad (6)$$

ただし, $i = 0, 1, \dots, u$ かつ $k = 1, 2, \dots, l_i$ である. また空集合に対する積は 1 であると約束する.

式 (5), (6) で与えられる M 関数はその時間区間に対して入れ子構造が存在するため, M 関数から得られる M 関数値は部分的な確率となる. ここで, 次の実現値が時間区間 $(t_{(i)}, t_{(i+1)})$ になる確率は次式のように導出できる.

$$\begin{aligned} \Pr\{X_{n+1} \in (t_{(i)}, t_{(i+1)})\} &= M_{X_{n+1}}(t_{(i)}, t_{(i+1)}) + \sum_{k=1}^{l_i} M_{X_{n+1}}(c_k^i, t_{(i+1)}) \\ &= \frac{1}{n+1} \prod_{\{r:c_{(r)} < t_{(i+1)}\}} \frac{\tilde{n}_{c_{(r)}} + 1}{\tilde{n}_{c_{(r)}}}, \quad i = 0, 1, \dots, u. \end{aligned} \quad (7)$$

なお, 式 (7) は, $i = 0, 1, \dots, u-1$ に対しては

$$\Pr\{X_{n+1} \in (t_{(i)}, t_{(i+1)})\} = M_{X_{n+1}}(t_{(i+1)}, t_{(i+2)}) \quad (8)$$

と書くことができる.

さらに, 式 (7) を用いて予測信頼度関数を導出する. X_{n+1} に対する予測信頼度関数は $S_{X_{n+1}}(t) = \Pr\{X_{n+1} > t\}$, ($t \geq 0$) と書くことができる. また, 予測信頼度関数の上限, 下限をそれぞれ $\bar{S}_{X_{n+1}}(t)$, $\underline{S}_{X_{n+1}}(t)$ と書くこととする. このとき, 予測信頼度関数の上下限は障害発生時刻 $t_{(i)}$ においては一致し,

$$\bar{S}_{X_{n+1}}(t_{(i)}) = \underline{S}_{X_{n+1}}(t_{(i)}) = \sum_{j=i}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\}, \quad i = 0, 1, \dots, u \quad (9)$$

となる.

時間区間 $(t_{(i)}, t_{(i+1)})$ の予測信頼度関数の上限を考えるときには, 時間区間 $(t_{(i)}, t_{(i+1)})$ に入れ子として含まれる M 関数, $M_{X_{n+1}}(c_k^i, t_{(i+1)})$, $k = 1, 2, \dots, l_i$, についてその時間区間 $(c_k^i, t_{(i+1)})$ に発生する可能性のある障害発生の時刻をその区間の上限である右端点, つまり $t_{(i+1)}$ へ移動すればよい. このとき, X_{n+1} に対する予測信頼度関数の上限は $i = 0, 1, \dots, u$ に対して

$$\bar{S}_{X_{n+1}}(t) = \bar{S}_{X_{n+1}}(t_{(i)}), \quad t \in [t_{(i)}, t_{(i+1)}) \quad (10)$$

となる. つまり, 予測信頼度関数の上限は各障害時刻 $t_{(i)}$ において $\Pr\{X_{n+1} \in (t_{(i-1)}, t_{(i)})\}$ だけ減少する階段関数となり, $t \in [0, t_{(1)})$ では $\bar{S}_{X_{n+1}}(t) = 1$, $t \in [t_{(u)}, \infty)$ に対しては正の値で一定となる. さらに, 時間区間 $(t_{(i)}, t_{(i+1)})$ で発生した打ち切りは, 同区間での予測信頼度関数の上限には影響を与えず, 時刻 $t_{(i+1)}$ 以降での減少量にのみ影響する. また, 各障害時刻 $t_{(i)}$ において, 予測信頼度関数の上限は右連続関数であることにも注意を要する.

同様に, 時間区間 $(t_{(i)}, t_{(i+1)})$ における予測信頼度関数の下限についても考える. 下限については, 時間区間 $(t_{(i)}, t_{(i+1)})$ に入れ子構造として含まれる M 関数, $M_{X_{n+1}}(c_k^i, t_{(i+1)})$, $k = 1, 2, \dots, l_i$ に対して, その時間区間 $(c_k^i, t_{(i+1)})$ に発生する可能性のある障害発生時刻がその区間の下限である左端点, すなわち c_k^i であるとして考えればよい. したがって, X_{n+1} に対する予測信頼度関数の下限は

$$\underline{S}_{X_{n+1}}(t) = \sum_{j=i+1}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} + \sum_{\{k:c_k^i \geq t\}} M_{X_{n+1}}(c_k^i, t_{(i+1)}), \quad t \in (t_{(i)}, t_{(i+1)}) \quad (11)$$

となる. なお, 数値計算の簡単化のために, すでに予測信頼度関数の上限が得られているのであればそれを利用し,

$$\underline{S}_{X_{n+1}}(t) = \bar{S}_{X_{n+1}}(t_{(i+1)}) + \sum_{\{k:c_k^i \geq t\}} M_{X_{n+1}}(c_k^i, t_{(i+1)}), \quad t \in (t_{(i)}, t_{(i+1)}) \quad (12)$$

とすることで導出も可能である。式 (11) で与えられる予測信頼度関数の下限もまた階段関数であるが、下限は障害発生時刻だけでなく右側打ち切り時刻においても減少する。また各障害時刻および右側打ち切り時刻 $x_{(i)}$ においては左連続関数である。

4. 適応的ソフトウェア若化スケジュール

ここでは 3. で導出した右側打ち切りデータを用いた予測信頼度関数の上下限を利用してシステムアベイラビリティの上下限を導出する。さらに、 $n+1$ 番目のシステム障害に対するシステムアベイラビリティの上下限をそれぞれ最大化する最適なソフトウェア若化スケジュールについて議論する。これにより、オンラインでシステム障害時間と右側打ち切り時間データを取得しながら次のシステム障害に対して適応的に次の最適若化スケジュールを導出できるようになる。

4.1 システムアベイラビリティの上限

まず、式 (10) で与えられた予測信頼度関数の上限を用いてシステムアベイラビリティの上限を導出し、 X_{n+1} に対してシステムアベイラビリティを最大にする最適若化スケジュールは既に観測された u 個の障害発生時刻の直前、すなわち $t_{(i)}^-$, $i = 1, 2, \dots, u$, のひとつとなることを示す。

今、 X_{n+1} に対するシステムアベイラビリティの上限を $\bar{A}_{X_{n+1}}(t_0)$ で定義する。式 (10) で与えられた予測信頼度関数の上限を式 (2) のシステムアベイラビリティへ代入することで、システムアベイラビリティの上限に関する次の補助定理を得る。

補助定理 4.1. ノンパラメトリック予測推論に基づいた X_{n+1} に対するシステムアベイラビリティの上限は次式によって与えられる。

$$\bar{A}_{X_{n+1}}(t_0) = \frac{\Phi_U(t_0)}{\Psi_U(t_0)}, \quad t_0 \in [t_{(i)}, t_{(i+1)}), \quad i = 0, 1, \dots, u. \quad (13)$$

ただし、

$$\Phi_U(t_0) = \mu_0 + t_0 \sum_{j=i}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} + \sum_{j=1}^{i-1} t_{(j+1)} \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\}, \quad (14)$$

$$\Psi_U(t_0) = \mu_0 + \mu_a + (\mu_c - \mu_a + t_0) \sum_{j=i}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\}$$

$$+ \sum_{j=1}^{i-1} t_{(j+1)} \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\}. \quad (15)$$

証明 X_{n+1} に関する予測信頼度関数の上限 $\bar{S}_{X_{n+1}}(t_0)$ が階段関数であり、かつ、 $t_{(i)}$, $i = 1, 2, \dots, u$ において右連続関数であるという性質を利用すると、 $t_0 \in [t_{(i)}, t_{(i+1)})$, $i = 0, 1, \dots, u$ に対して

$$\begin{aligned} \int_0^{t_0} \bar{S}_{X_{n+1}}(x) dx &= \int_0^{t_{(i)}} \bar{S}_{X_{n+1}}(x) dx + \int_{t_{(i)}}^{t_0} \bar{S}_{X_{n+1}}(x) dx \\ &= t_0 \bar{S}_{n+1}(t_{(i)}) + \sum_{j=0}^{i-1} t_{(j+1)} \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} \end{aligned} \quad (16)$$

が得られる。式 (10) および式 (16) を式 (2) に代入すれば式 (13) が得られる。 \square

補助定理 4.2. システムアベイラビリティの上限 $\bar{A}_{X_{n+1}}(t_0)$ は $t_0 \in (t_{(i)}, t_{(i+1)})$, $i = 0, 1, \dots, u$ において連続な単調増加関数である。更に、 $\bar{A}_{X_{n+1}}(t_0)$ は $t_{(i)}$, $i = 0, 1, \dots, u$ において右連続であり、 $t_{(i)}^-$, $i = 1, 2, \dots, u$ において局所的に最大化される。

証明 予測信頼度関数 $\bar{S}_{X_{n+1}}(t_0)$ は $t_0 \in (t_{(i)}, t_{(i+1)})$, $i = 0, 1, \dots, u-1$ において連続であるので、システムアベイラビリティの上限 $\bar{A}_{X_{n+1}}(t_0)$ も連続関数である。また、 $\bar{A}_{X_{n+1}}(t_0)$ は $t_0 \in (t_{(i)}, t_{(i+1)})$, $i = 0, 1, \dots, u-1$ において単調増加関数である。さらに、

$$\lim_{\epsilon \rightarrow 0} \bar{A}_{X_{n+1}}(t_{(i)} + \epsilon) = \bar{A}_{X_{n+1}}(t_{(i)}) \quad (17)$$

であるので、 $\bar{A}_{X_{n+1}}(t_0)$ は $t_0 = t_{(i)}$, $i = 0, 1, \dots, u$ において右連続である。一方で、

$$\lim_{\epsilon \rightarrow 0} \bar{A}_{X_{n+1}}(t_{(i)} - \epsilon) = \frac{\lim_{\epsilon \rightarrow 0} \Phi_U(t_{(i)} - \epsilon)}{\lim_{\epsilon \rightarrow 0} \Psi_U(t_{(i)} - \epsilon)} \quad (18)$$

とすると、

$$\begin{aligned} &\lim_{\epsilon \rightarrow 0} \Phi_U(t_{(i)} - \epsilon) \\ &= \mu_0 + t_{(i)} \sum_{j=i}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} + \sum_{j=1}^{i-1} t_{(j+1)} \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} \\ &= \Phi_U(t_{(i)}), \end{aligned} \quad (19)$$

$$\begin{aligned} &\lim_{\epsilon \rightarrow 0} \Psi_U(t_{(i)} - \epsilon) \\ &= \mu_0 + \mu_a + (\mu_c - \mu_a) \sum_{j=i-1}^u \Pr\{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} \end{aligned}$$

$$+ t_{(i)} \sum_{j=i}^u \Pr \{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} + \sum_{j=1}^{i-1} t_{(j+1)} \Pr \{X_{n+1} \in (t_{(j)}, t_{(j+1)})\} \quad (20)$$

である。ここで、 $\mu_c < \mu_a$ の条件により、

$$\lim_{\epsilon \rightarrow 0} \Psi_U(t_{(i)} - \epsilon) < \Psi_U(t_{(i)}) \quad (21)$$

であることから

$$\lim_{\epsilon \rightarrow 0} \bar{A}_{X_{n+1}}(t_{(i)} - \epsilon) > \bar{A}_{X_{n+1}}(t_{(i)}) \quad (22)$$

である。したがって、 $\bar{A}_{X_{n+1}}(t_0)$ は $t_{(i)}$, $i = 0, 1, \dots, u$ において右連続であり、 $t_{(i)}^-$, $i = 1, 2, \dots, u$ で局所的に最大化される。□

補助定理 4.2. より、次の定理が得られる。

定理 4.3. $t_0 \in (0, t_{(u)}]$ においてシステムアベイラビリティの上限、 $\bar{A}_{X_{n+1}}(t_0)$ の最大値は $t_{(i)}^-$, $i = 1, 2, \dots, u$ のいずれかの点で与えられる。

X_{n+1} に対してソフトウェア若化を実施しない場合、すなわち $t_0 = \infty$ のとき、 $t_{(u+1)} = \infty$ を仮定すれば明らかにシステムアベイラビリティの上限は 1 となる。つまり、 $t_{(u+1)} = \infty$ が仮定される状況では、システムアベイラビリティを最大化するためにはソフトウェアの予防的若化を一切行わないことが最適である。よって、 $t_0^* \rightarrow \infty$ である。

一方で $t_{(u+1)}$ に既知の上限 r が存在すると仮定できるのであれば、定理 4.3. で与えられる最大アベイラビリティ $\bar{A}_{X_{n+1}}(t_{(i^*)}^-)$ と $\bar{A}_{X_{n+1}}(r^-)$ を比較することで最適若化スケジュールを決定できる。つまり、 $\bar{A}_{X_{n+1}}(t_{(i^*)}^-) \geq \bar{A}_{X_{n+1}}(r^-)$ なら $t_0^* = t_{(i^*)}^-$ が最適ソフトウェア若化スケジュールとなり、 $\bar{A}_{X_{n+1}}(t_{(i^*)}^-) < \bar{A}_{X_{n+1}}(r^-)$ ならソフトウェア若化を実施しないことが最適な方策となる。さらに、 $t_{(u+1)}$ に未知ではあるが上限 r が存在すると仮定できる場合には、 $\bar{A}_{X_{n+1}}(t_{(i^*)}^-) = \bar{A}_{X_{n+1}}(r^-)$ を満たす r をしきい値 r^* とし、想定可能な X_{n+1} の上限 r が $r \leq r^*$ なら最適ソフトウェア若化スケジュールは $t_0^* = t_{(i^*)}^-$ となり、 $r > r^*$ ならソフトウェア若化を実施しないことが最適な方策となる。

4.2 システムアベイラビリティの下限

次に式 (11) で定式化した予測信頼度関数の下限を用いてシステムアベイラビリティの下限を導出する。さらに、 X_{n+1} に対するシステムアベイラビリティの下限を最大にする最適若化スケジュールは既に観測された n 個の障害または打ち切り時刻のひとつとなることを示す。

X_{n+1} に対するシステムアベイラビリティの下限を $\underline{A}_{X_{n+1}}(t_0)$ と書くことにする. 式 (11) で与えられた予測信頼度関数の下限を式 (2) のシステムアベイラビリティへ代入すると, 次の補助定理が得られる.

補助定理 4.4. X_{n+1} に対するノンパラメトリック予測推論に基づいたシステムアベイラビリティの下限は

$$\underline{A}_{X_{n+1}}(t_0) = \frac{\Phi_L(t_0)}{\Psi_L(t_0)}, \quad t_0 \in (x_{(i)}, x_{(i+1)}], \quad i = 0, 1, \dots, n, \quad (23)$$

によって与えられる. ただし,

$$\Phi_L(t_0) = \mu_0 + \sum_{j=0}^{i-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) + (t_0 - x_{(i)}) \underline{S}_{X_{n+1}}(x_{(i+1)}), \quad (24)$$

$$\begin{aligned} \Psi_L(t_0) &= \mu_0 + \mu_a + (\mu_c - \mu_a) \underline{S}_{X_{n+1}}(t_0) + \sum_{j=0}^{i-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) \\ &\quad + (t_0 - x_{(i)}) \underline{S}_{X_{n+1}}(x_{(i+1)}), \end{aligned} \quad (25)$$

である.

証明 X_{n+1} の予測信頼度関数の下限 $\underline{S}_{X_{n+1}}(t_0)$ が階段関数であり, かつ, $x_{(i)}, i = 1, 2, \dots, n$ において左連続であるという性質を利用すると, $t_0 \in (x_{(i)}, x_{(i+1)}], i = 0, 1, \dots, n$ に対して

$$\begin{aligned} \int_0^{t_0} \underline{S}_{X_{n+1}}(x) dx &= \int_0^{x_{(i)}} \underline{S}_{X_{n+1}}(x) dx + \int_{x_{(i)}}^{t_0} \underline{S}_{X_{n+1}}(x) dx \\ &= \sum_{j=0}^{i-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) + (t_0 - x_{(i)}) \underline{S}_{X_{n+1}}(x_{(i+1)}) \end{aligned} \quad (26)$$

が得られる. 式 (26) を式 (2) に代入すれば式 (23) が得られる. \square

明らかに $t_0 \in (x_{(n)}, \infty]$ に対して $\underline{S}_{X_{n+1}}(t_0) = 0$ であるので, システムアベイラビリティの下限は

$$\underline{A}_{X_{n+1}}(t_0) = \frac{\mu_0 + \sum_{j=0}^{n-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)})}{\mu_0 + \mu_a + \sum_{j=0}^{n-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)})} \quad (27)$$

で一定となる. さらに,

$$\underline{A}_{X_{n+1}}(x_{(n)}) = \frac{\mu_0 + \sum_{j=0}^{n-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)})}{\mu_0 + \mu_a + (\mu_c - \mu_a) \underline{S}_{X_{n+1}}(x_{(n)}) + \sum_{j=0}^{n-1} (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)})} \quad (28)$$

であるので, $\mu_c < \mu_a$ の条件から,

$$\underline{A}_{X_{n+1}}(x_{(n)}) > \underline{A}_{X_{n+1}}(t_0), \quad t_0 \in (x_{(n)}, \infty] \quad (29)$$

である. したがって, システムアベイラビリティの下限に関しては, $t_0 \in (x_{(n)}, \infty]$ は最適なソフトウェア若化スケジュールになり得ない.

補助定理 4.5. システムアベイラビリティの下限 $\underline{A}_{X_{n+1}}(t_0)$ は $t_0 \in (x_{(i)}, x_{(i+1)})$, $i = 0, 1, \dots, n-1$ において連続な単調増加関数である. 更に, $\underline{A}_{X_{n+1}}(t_0)$ は $x_{(i)}$, $i = 1, 2, \dots, n$ において左連続であり, $x_{(i)}$, $i = 1, 2, \dots, n$ において局所的に最大化される.

証明 式 (23) より, $t_0 \in (x_{(i)}, x_{(i+1)})$, $i = 0, 1, \dots, n-1$ において, $\underline{S}_{X_{n+1}}(t_0)$ は連続関数である. また明らかに $\underline{A}_{X_{n+1}}(t_0)$ は, $t_0 \in (x_{(i)}, x_{(i+1)})$, $i = 0, 1, \dots, n-1$ において単調増加関数である. 式 (23) において, $t_0 = x_{(i+1)}$, $i = 0, 1, \dots, n-1$ とすると,

$$\Phi_L(x_{(i+1)}) = \mu_0 + \sum_{j=0}^i (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}), \quad (30)$$

$$\begin{aligned} \Psi_L(x_{(i+1)}) &= \mu_0 + \mu_a + (\mu_c - \mu_a) \underline{S}_{X_{n+1}}(x_{(i+1)}) \\ &\quad + \sum_{j=0}^i (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) \end{aligned} \quad (31)$$

である. さらに

$$\lim_{\epsilon \rightarrow 0} \underline{A}_{X_{n+1}}(x_{(i+1)} + \epsilon) = \frac{\lim_{\epsilon \rightarrow 0} \Phi_L(x_{(i+1)} + \epsilon)}{\lim_{\epsilon \rightarrow 0} \Psi_L(x_{(i+1)} + \epsilon)} \quad (32)$$

とすると

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \Phi_L(x_{(i+1)} + \epsilon) &= \mu_0 + \sum_{j=0}^i (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) \\ &= \Phi_L(x_{(i+1)}), \end{aligned} \quad (33)$$

である. また, $\lim_{\epsilon \rightarrow 0} \underline{S}_{X_{n+1}}(x_{(i+1)} + \epsilon) = \underline{S}_{X_{n+1}}(x_{(i+2)})$ であり, かつ $\mu_c < \mu_a$ という条件に注意すれば

$$\begin{aligned} &\lim_{\epsilon \rightarrow 0} \Psi_L(x_{(i+1)} + \epsilon) \\ &= \mu_0 + \mu_a + (\mu_c - \mu_a) \underline{S}_{X_{n+1}}(x_{(i+2)}) + \sum_{j=0}^i (x_{(j+1)} - x_{(j)}) \underline{S}_{X_{n+1}}(x_{(j+1)}) \\ &> \Psi_L(x_{(i+1)}) \end{aligned} \quad (34)$$

なる関係が得られる。よって

$$\lim_{\epsilon \rightarrow 0} \underline{A}_{X_{n+1}}(x_{(i+1)} + \epsilon) < \underline{A}_{X_{n+1}}(x_{(i+1)}) \quad (35)$$

が成立する。一方で,

$$\lim_{\epsilon \rightarrow 0} \underline{A}_{X_{n+1}}(x_{(i+1)} - \epsilon) = \underline{A}_{X_{n+1}}(x_{(i+1)}) \quad (36)$$

である。したがって, $\underline{A}_{X_{n+1}}(t_0)$ は $t_0 = x_{(i)}$, $i = 1, 2, \dots, n$ において左連続であり, $x_{(i)}$, $i = 1, 2, \dots, n$ で局所的に最大化される。□

定理 4.6. システムアベイラビリティの下限, $\underline{A}_{X_{n+1}}(t_0)$ の最大値は, $x_{(i)}$, $i = 1, 2, \dots, n$ のいずれかの点で与えられる。

4.3 数値例

ここでは本論文で提案した右側打ち切り時間を含んだシステム障害時間データからノンパラメトリック予測推論によって最適ソフトウェア若化スケジュールを推定する数値例を示す。システムアベイラビリティの上限, 下限を最大化する最適ソフトウェア若化スケジュールをそれぞれ $T_{\text{up},n+1}^* = \text{argmax} \bar{A}_{X_{n+1}}(t_0^-)$, $T_{\text{low},n+1}^* = \text{argmax} \underline{A}_{X_{n+1}}(t_0)$ と書くことにする。システム障害時間分布は未知であるが, $u = 4$ 個のシステム障害時刻の順序標本 (825, 1598, 3737, 4589) および, $v = 4$ 個の右側打ち切り時刻データの順序標本 (1100, 2150, 2550, 5050) が得られている状況を考える。その他のパラメータは $\mu_0 = 240$, $\mu_a = 0.50$, $\mu_c = 0.16$ と設定する。

表 1 には式 (5), (6) から得られる M 関数値と, 式 (7) によって計算される X_{n+1} に対する確率 $\Pr\{X_{n+1} \in (t_{(i)}, t_{(i+1)})\}$ を示す。表 1 において M 関数値の時間間隔は右側打ち切り時刻に関するものが, システム障害時刻間隔に入れ子構造となって存在していることに注意を要する。たとえば, システム障害時刻間隔 $(t_{(2)}, t_{(3)}) = (1598, 3737)$ にはその間に発生した 2 つの打ち切り時刻から始まる時間間隔 $(c_1^2, t_{(3)}) = (2150, 3737)$, $(c_2^2, t_{(3)}) = (2550, 3737)$ の M 関数が入れ子として存在している。また式 (7) により, $(t_{(2)}, t_{(3)}) = (1598, 3737)$ に対する確率は, $(t_{(2)}, t_{(3)}) = (1598, 3737)$ の M 関数値と, そこに入れ子として含まれる M 関数値を合計したものと一致する。つまり, $\Pr\{X_{n+1} \in (t_{(2)}, t_{(3)})\} = 8/63 + 8/315 + 4/105 = 4/21$ である。

図 2 にはそのような $n = 8$ 個の障害および右側打ち切りデータが得られている状況で, X_{n+1} に対するノンパラメトリック予測推論に基づいた予測信頼度関数の上下限の振る舞いを示す。図 2 より, 予測信頼度関数の上限は障害時刻のみで減少する階段関数であるのに対して, 下限

表1 M -関数値および $\Pr\{X_{n+1} \in (t_{(i)}, t_{(i+1)})\}$

Interval	M -function	Probabilitiy
$(t_{(0)}, t_{(1)})$	1/9	1/9
$(t_{(1)}, t_{(2)})$	1/9	8/63
$(c_1^1, t_{(2)})$	1/63	-
$(t_{(2)}, t_{(3)})$	8/63	4/21
$(c_1^2, t_{(3)})$	8/315	-
$(c_2^2, t_{(3)})$	4/105	-
$(t_{(3)}, t_{(4)})$	4/21	4/21
$(t_{(4)}, \infty)$	4/21	8/21
(c_1^4, ∞)	4/21	-

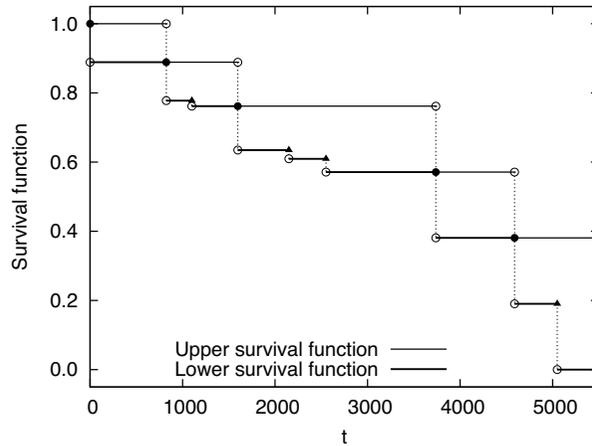


図2 X_{n+1} に対する予測信頼度関数

は障害時刻と右側打ち切り時刻の両方で減少する階段関数であることが確認できる。また、各障害間で右側打ち切りが観測された場合には予測信頼度関数の減少幅が大きくなるとともに、予測信頼度関数の上下限の差が徐々に大きくなることに注意を要する。

表2には8個の障害および右側打ち切り時刻データとシステムアベイラビリティの上下限を示し、図3にはシステムアベイラビリティの上下限の振る舞いを示す。表2および

表 2 8 個の障害及び右側打ち切り時刻データとシステムアベイラビリティの上下限

	$t_{(1)}$	$c_{(1)}$	$t_{(2)}$	$c_{(2)}$	$c_{(3)}$	$t_{(3)}$	$t_{(4)}$	$c_{(4)}$
$t_{(i)}, c_{(i)}$ or $x_{(i)}$	825	1100	1598	2150	2550	3737	4589	5050
$\bar{A}_{X_{n+1}}(t_{(i)}^-)$	0.999850	-	0.999887	-	-	0.999929	0.999921	-
$\underline{A}_{X_{n+1}}(x_{(i)})$	0.999797	0.999802	0.999846	0.999852	0.999865	0.999892	0.999883	0.999866

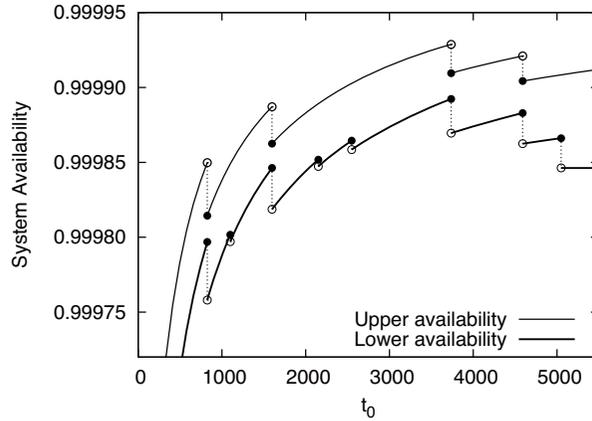


図 3 X_{n+1} に対するシステムアベイラビリティ上下限の振る舞い

図 3 より，システムアベイラビリティの上下限をそれぞれ最大にするソフトウェア若化スケジュールは $T_{\text{up},n+1}^* = T_{\text{low},n+1}^* = 3737$ となり，最大システムアベイラビリティは $\bar{A}_{X_{n+1}}(3737^-) = 0.999929$ ， $\underline{A}_{X_{n+1}}(3737) = 0.999892$ と推定される．なお，システムアベイラビリティの上限に関するしきい値は $r^* = 8083$ となる．

5. シミュレーション実験

ここでは本論文で提案した右側打ち切りデータに対するノンパラメトリック予測推論に基づいたソフトウェア若化スケジュールの推定に関する 2 種類のシミュレーション実験を行う．提案手法によって推定された予防若化スケジュールの推定値と理論的な最適解との比較を行い，提案手法の有効性について検証する．以下では障害発生時間 X がワイブル分布

$$F_f(t) = 1 - e^{-(t/\theta)^\gamma}, \quad \gamma > 0, \theta > 0 \tag{37}$$

表3 ケース設定及び理論的な最適ソフトウェア若化スケジュールと最大アベイラビリティ

Case	γ	θ	λ_f	SD[X]	μ_0	μ_a	μ_c	t_0^*	$A(t_0^*)$
Case I	1.5	2215.46	2000.00	1357.93	240.00	0.50	0.16	1816.62	0.999792
Case II	2.0	2256.76	2000.00	1045.45	240.00	0.50	0.16	1359.67	0.999819
Case III	4.0	2206.53	2000.00	561.09	240.00	0.50	0.16	1316.40	0.999869

に従うものと仮定する。ここで、 γ , θ はそれぞれ形状パラメータ、尺度パラメータである。表3には3種類のケース設定および障害発生時間分布とそのパラメータが既知である場合に得られる理論的な最適ソフトウェア若化スケジュールと最大アベイラビリティを示す。表3から、各ケースにおいて平均障害発生時間は同じであるが、 γ が大きくなるに従って、障害発生時間の標準偏差 SD[X] は小さくなり、同時に最適なソフトウェア若化スケジュールは短く、最大アベイラビリティは大きくなることを確認できる。

5.1 ランダム打ち切りデータに対するオンライン学習シミュレーション実験

まず、障害時間データとランダム打ち切りデータが逐次得られる状況を想定したシミュレーション実験を行う。これは、例えば同種類のソフトウェアシステムが複数稼働しており、これらの各ソフトウェアシステムの障害や予防若化時間に関するデータが逐次得られるような状況を考慮している。

ランダム打ち切りは連続時間分布 $G(t)$ に従って観測されるとする。このとき、障害データにランダム打ち切りデータが含まれる割合、すなわち打ち切り確率 q は次式によって与えられる [38].

$$q = \int_0^{\infty} S(t) dG(t). \quad (38)$$

つまり、逐次得られる新たなデータは、確率 q で右側打ち切りデータであり、確率 $1 - q$ で障害データである。ここでは、打ち切り時間分布 $G(t)$ に対して平均 ν の指数分布

$$G(t) = 1 - e^{-t/\nu}, \quad \nu > 0 \quad (39)$$

を仮定する。表4には各ケースに対する打ち切り確率と打ち切り時間分布のパラメータを示す。なお、特別な場合として $q = 0$ は打ち切りが存在しない完全データを意味するものとする。

以下では式 (37) と式 (39) によって与えられる障害時間分布および打ち切り時間分布に基づいた疑似乱数を用いてモンテカルロシミュレーションを行う。なお、現実的な想定として、障害

表4 打ち切り確率 (q) と打ち切り時間分布のパラメータ (ν)

q	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Case-I	18522.04	8503.26	5148.83	3458.11	2430.20	1730.29	1213.01	801.90	444.11
Case-II	18708.46	8688.01	5331.60	3638.45	2607.45	1903.45	1380.38	960.15	584.50
Case-III	18903.38	8883.31	5527.30	3834.57	2804.01	2100.42	1577.68	1157.44	780.24

時間と打ち切り時間のペアのうち小さい方のデータのみが観測データとして利用できるものとする。このシミュレーション実験によりソフトウェアの最適若化スケジュールと最大アベイラビリティの推定値が理論的な最適解に収束する様子について調査する。

図4はケースIIの $q = 0.2$ に対する最適若化スケジュールと最大アベイラビリティ推定値の漸近的な振る舞いを示す。ここで横軸のデータ数は障害データ数 (u) と打ち切りデータ数 (v) の合計 (n) を意味している。また理論的な最適解は以降の各図中においても横軸と平行な点線によって示される。図4より、信頼度関数の上限と下限をそれぞれ考慮した最適若化スケジュールはほとんどの場合において一致していることが読み取れる。また $n = 30$ 程度のデータが得られれば上限、下限ともに理論的な最適解に収束している様子を読み取ることができる。

次に図4で示したシミュレーション実験をケースIIの複数の q についてそれぞれ1000回ずつ繰り返して得られた結果について考察を行う。図5および図6には1000回のシミュレーション実験によって得られた最適若化スケジュールと最大アベイラビリティの推定値に対する平均値および中央値の漸近的振る舞いをそれぞれ示す。図5、図6より、上限、下限の推定値に対する平均値および中央値はいずれの場合にも理論的な最適解に収束する様子が確認できる。また q が小さいほど、すなわち打ち切りデータが含まれる割合が小さいほど、推定値の収束は速くなることも読み取ることができる。さらに、最適若化スケジュールの推定値に関しては平均値、中央値ともに n が小さい時には理論的な最適解よりも大きく推定される傾向があり、 n が大きくなればともに理論的な最適解に収束していることが分かる。一方で、最大アベイラビリティの推定値は n が小さい時には上限と下限の差が大きいものの、 n が大きくなるにつれてその差は小さくなり、理論的な最大アベイラビリティよりも僅かに大きな値に収束する傾向がある。

図7および8には1000回のシミュレーション実験における推定値の標準偏差と平均絶対誤差率 (Mean Absolute Percentage Error: MAPE) の漸近的振る舞いを示す。ここで平均絶対

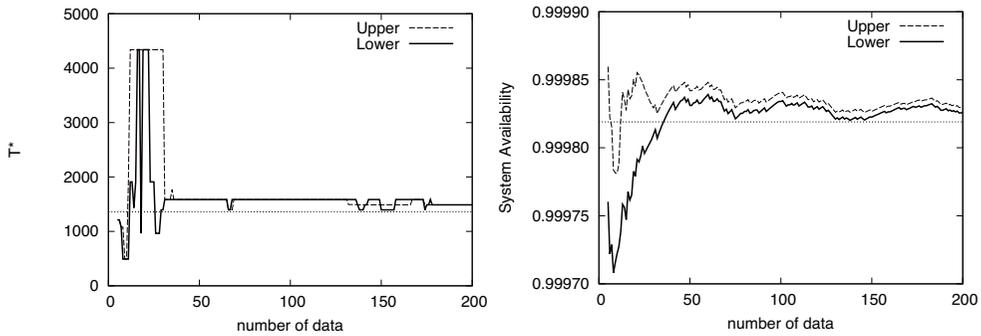


図4 推定値に関する漸近的性質 ($q = 0.2$)

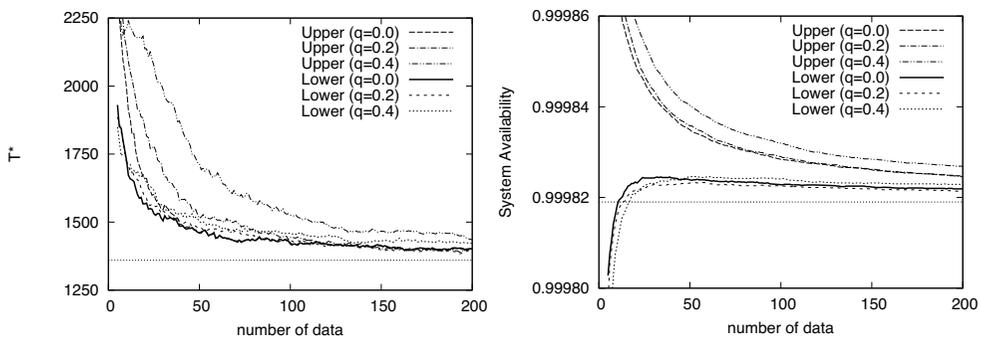


図5 推定値の平均値に関する漸近的性質

誤差率は

$$MAPE = \frac{1}{m} \sum_{i=1}^m \left| \frac{Y_i - Y^*}{\bar{Y}_i} \right| \times 100\% \quad (40)$$

$$\bar{Y}_i = \frac{1}{m} \sum_{i=1}^m |Y_i| \quad (41)$$

で定義され、 m はシミュレーション回数、 Y_i は各シミュレーションで得られた推定値、 Y^* は理論的な最適解である。図7および図8より、推定値の標準偏差と平均絶対誤差率は n が大きくなるにつれて小さくなっていく様子が確認できる。また、それらの減少速度も平均値や中央値と同様に q が小さいほど速いことが読み取れる。

表5および6には各ケースの $n = 10, n = 100, n = 200$ と $q = 0.0, q = 0.2, q = 0.4$ の組合せについてそれぞれ1000回のシミュレーション実験によって得られた上限、下限の結果を示す。表5および表6より、最適若化スケジュール推定値の平均値と中央値は上限、下限とも

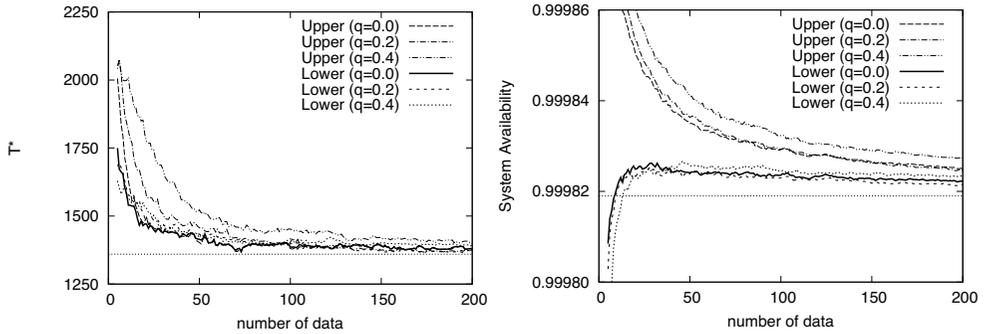


図6 中央値に関する漸近的性質

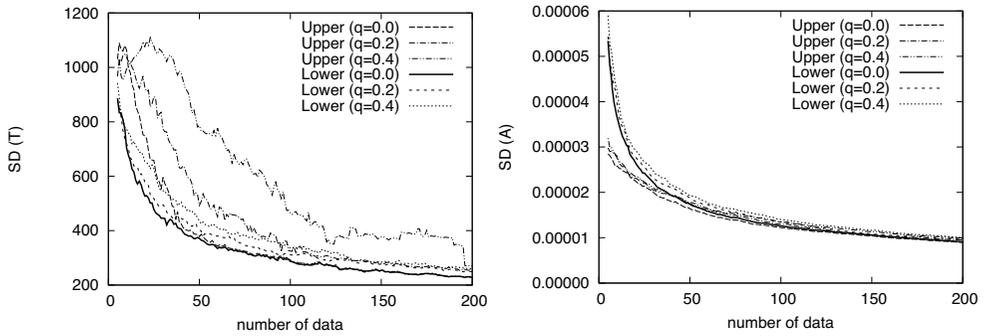


図7 推定値の標準偏差に関する漸近的性質

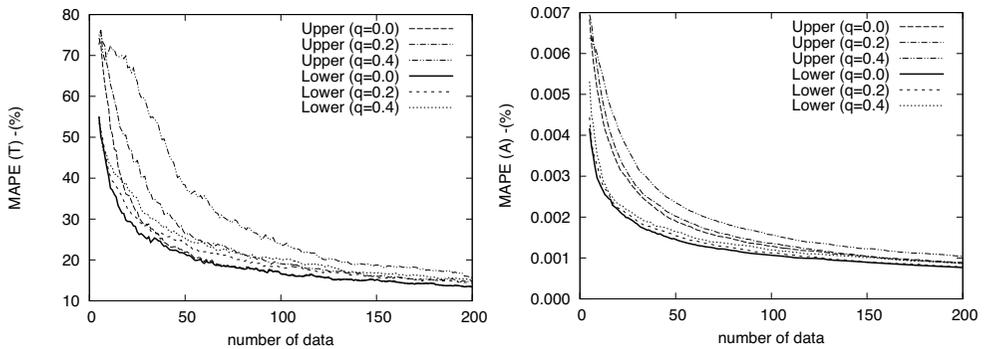


図8 推定値の平均絶対誤差率に関する漸近的性質

にすべての場合で理論的な最適解よりも大きくなっていることが読み取れる。またその標準偏差や平均絶対誤差率は下限についてはすべての場合でデータ数の増加とともに小さくなっていることが分かる。上限についても標準偏差と平均絶対誤差率は $q = 0.0$ でデータ数の増加とともに小さくなっているが、 $q = 0.2$ と $q = 0.4$ では逆に大きくなる場合も存在している。

最大アベイラビリティ推定値の平均値と中央値については、上限のケース I とケース II では理論的な最大アベイラビリティよりも大きく、ケース III では小さくなっている。下限はケース III では理論的な最大アベイラビリティよりも小さく、ケース I とケース II では n が大きくなれば理論的な最大アベイラビリティよりも大きくなることが分かる。アベイラビリティ推定値の標準偏差や平均絶対誤差率はデータ数の増加とともに小さくなる。これは障害、打ち切りデータを逐次得ることでより高精度に推定が可能になるという本研究で提案した手法の有効性を示すものである。また打ち切り確率 q が小さいほど標準偏差や平均絶対誤差率が小さくなる様子も確認できる。

さらに表 5 と表 6 で最大アベイラビリティの平均絶対誤差率を比較すれば、下限の平均絶対誤差率がすべての場合において小さくなっている様子を読み取ることができる。これは図 4 に示したとおり、上限と下限の最適若化スケジュールの推定値は多くの場合に一致するが、もしもこれらの推定値が異なった場合にも下限のスケジュールを採用すればよいことを意味している。

5.2 適応的スケジューリングに対するシミュレーション実験

次に単一のソフトウェアシステムが稼働しており、このシステム運用段階において障害時間データを収集しながら適応的に最適若化スケジューリングを実施するような状況を想定したシミュレーション実験を行う。このような状況において本研究で提案した若化方を厳密に適用した場合、定理 4.3. および定理 4.6. で示したとおり、予防的ソフトウェア若化は過去に観測された障害時刻の直前または同時刻に実施されることに注意を要する。これはすなわち障害時刻とタイになる打ち切りデータしか得られないことを意味している。便宜上、Coolen and Yan [30] などの取り扱いと同様に、障害時刻とタイになる打ち切り時刻は、障害時刻の直後に配置することにする。

図 9 にはケース II に対する適応的な最適若化スケジュールと最大アベイラビリティ推定値の漸近的な振る舞いを示す。図 9 より、 $n = 30$ 程度のデータが得られれば、上限、下限ともに理論的な最適解に収束していることが分かる。

図 10~12 にはケース I~III に対して図 9 で示したシミュレーション実験をそれぞれ 1000 回

表5 シミュレーション実験結果 (上限)

	$q = 0.0$		$q = 0.2$		$q = 0.4$	
	$T_{\text{up},n+1}^{-*}$	$A_{\text{up},n+1}$	$T_{\text{up},n+1}^{-*}$	$A_{\text{up},n+1}$	$T_{\text{up},n+1}^{-*}$	$A_{\text{up},n+1}$
Case I : $n = 10$ ($t_0^* = 1816.62, A(t_0^*) = 0.999792$)						
Mean	3083.24	0.9998563	2940.55	0.9998612	2844.35	0.9998673
Median	2556.32	0.9998609	2625.16	0.9998654	2700.55	0.9998720
SD	1785.33	3.06E-05	1563.79	3.44E-05	1354.77	3.77E-05
MAPE	87.18%	0.0066%	79.51%	0.0071%	71.58%	0.0078%
Case I : $n = 100$						
Mean	2172.93	0.9998047	2741.19	0.9998070	3478.60	0.9998141
Median	1891.55	0.9998046	2028.84	0.9998078	2784.56	0.9998140
SD	1208.42	1.44E-05	1885.76	1.48E-05	1932.64	1.61E-05
MAPE	37.19%	0.0016%	68.37%	0.0018%	101.21%	0.0023%
Case I : $n = 200$						
Mean	1983.95	0.9997999	2292.19	0.9998005	3065.87	0.9998043
Median	1866.94	0.9998004	1950.02	0.9998004	2204.15	0.9998049
SD	745.50	1.04E-05	1338.37	1.08E-05	1943.25	1.17E-05
MAPE	25.55%	0.0011%	40.76%	0.0011%	81.05%	0.0014%
Case II : $n = 10$ ($t_0^* = 1359.67, A(t_0^*) = 0.999819$)						
Mean	1990.79	0.9998662	2138.93	0.9998708	2211.38	0.9998757
Median	1675.74	0.9998682	1843.56	0.9998729	1998.61	0.9998780
SD	1062.90	2.58E-05	1036.51	2.76E-05	994.22	2.80E-05
MAPE	57.37%	0.0049%	66.09%	0.0053%	70.99%	0.0058%
Case II : $n = 100$						
Mean	1418.18	0.9998284	1443.14	0.9998290	1526.18	0.9998320
Median	1380.66	0.9998292	1404.27	0.9998293	1451.84	0.9998323
SD	291.58	1.22E-05	324.96	1.30E-05	465.96	1.35E-05
MAPE	16.86%	0.0013%	19.09%	0.0014%	23.75%	0.0016%
Case II : $n = 200$						
Mean	1401.79	0.9998247	1401.87	0.9998246	1435.71	0.9998269
Median	1379.90	0.9998250	1375.15	0.9998246	1397.79	0.9998274
SD	228.11	8.97E-06	257.51	9.53E-06	269.51	9.80E-06
MAPE	13.43%	0.0009%	14.93%	0.0009%	15.59%	0.0010%
Case III : $n = 10$ ($t_0^* = 1316.40, A(t_0^*) = 0.999869$)						
Mean	3083.24	0.9998563	2940.55	0.9998612	2844.35	0.9998673
Median	2556.32	0.9998609	2625.16	0.9998654	2700.55	0.9998720
SD	1785.33	3.06E-05	1563.79	3.44E-05	1354.77	3.77E-05
MAPE	87.18%	0.0066%	79.51%	0.0071%	71.58%	0.0078%
Case III : $n = 100$						
Mean	2172.93	0.9998047	2741.19	0.9998070	3478.60	0.9998141
Median	1891.55	0.9998046	2028.84	0.9998078	2784.56	0.9998140
SD	1208.42	1.44E-05	1885.76	1.48E-05	1932.64	1.61E-05
MAPE	37.19%	0.0016%	68.37%	0.0018%	101.21%	0.0023%
Case III : $n = 200$						
Mean	1983.95	0.9997999	2292.19	0.9998005	3065.87	0.9998043
Median	1866.94	0.9998004	1950.02	0.9998004	2204.15	0.9998049
SD	745.50	1.04E-05	1338.37	1.08E-05	1943.25	1.17E-05
MAPE	25.55%	0.0011%	40.76%	0.0011%	81.05%	0.0014%

表6 シミュレーション実験結果 (下限)

	$q = 0.0$		$q = 0.2$		$q = 0.4$	
	$T_{low,n+1}^*$	$A_{low,n+1}$	$T_{low,n+1}^*$	$A_{low,n+1}$	$T_{low,n+1}^*$	$A_{low,n+1}$
Case I : $n = 10$ ($t_0^* = 1816.62, A(t_0^*) = 0.999792$)						
Mean	2289.31	0.9997973	2206.69	0.9997931	2186.37	0.9997861
Median	1943.12	0.9998041	1861.92	0.9997978	1862.22	0.9997920
SD	1349.20	4.56E-05	1313.49	4.93E-05	1281.37	5.31E-05
MAPE	53.35%	0.0037%	52.23%	0.0039%	51.70%	0.0041%
Case I : $n = 100$						
Mean	1967.40	0.9997979	2009.91	0.9997978	2140.89	0.9997992
Median	1846.89	0.9997980	1821.84	0.9997985	1899.61	0.9997995
SD	673.50	1.51E-05	885.06	1.59E-05	1027.64	1.67E-05
MAPE	27.31%	0.0013%	31.87%	0.0014%	37.17%	0.0015%
Case I : $n = 200$						
Mean	1926.24	0.9997965	1948.24	0.9997962	2024.11	0.9997971
Median	1849.61	0.9997969	1834.06	0.9997961	1882.54	0.9997976
SD	523.55	1.07E-05	627.34	1.13E-05	762.54	1.21E-05
MAPE	22.66%	0.0009%	24.38%	0.0010%	29.66%	0.0011%
Case II : $n = 10$ ($t_0^* = 1359.67, A(t_0^*) = 0.999819$)						
Mean	1725.66	0.9998182	1716.49	0.9998159	1717.39	0.9998095
Median	1581.38	0.9998210	1599.33	0.9998202	1570.25	0.9998133
SD	703.34	3.67E-05	718.98	3.95E-05	773.71	4.24E-05
MAPE	40.52%	0.0029%	41.70%	0.0031%	43.96%	0.0034%
Case II : $n = 100$						
Mean	1423.54	0.9998228	1430.26	0.9998225	1459.44	0.9998241
Median	1386.74	0.9998236	1398.13	0.9998230	1416.39	0.9998245
SD	286.75	1.26E-05	311.20	1.35E-05	334.07	1.40E-05
MAPE	16.59%	0.0011%	18.23%	0.0011%	20.06%	0.0012%
Case II : $n = 200$						
Mean	1401.86	0.9998219	1395.03	0.9998213	1422.32	0.9998229
Median	1379.22	0.9998222	1372.41	0.9998214	1392.71	0.9998234
SD	227.65	9.10E-06	249.50	9.68E-06	257.98	9.99E-06
MAPE	13.40%	0.0008%	14.52%	0.0008%	15.14%	0.0009%
Case III : $n = 10$ ($t_0^* = 1316.40, A(t_0^*) = 0.999869$)						
Mean	2289.31	0.9997973	2206.69	0.9997931	2186.37	0.9997861
Median	1943.12	0.9998041	1861.92	0.9997978	1862.22	0.9997920
SD	1349.20	4.56E-05	1313.49	4.93E-05	1281.37	5.31E-05
MAPE	53.35%	0.0037%	52.23%	0.0039%	51.70%	0.0041%
Case III : $n = 100$						
Mean	1967.40	0.9997979	2009.91	0.9997978	2140.89	0.9997992
Median	1846.89	0.9997980	1821.84	0.9997985	1899.61	0.9997995
SD	673.50	1.51E-05	885.06	1.59E-05	1027.64	1.67E-05
MAPE	27.31%	0.0013%	31.87%	0.0014%	37.17%	0.0015%
Case III : $n = 200$						
Mean	1926.24	0.9997965	1948.24	0.9997962	2024.11	0.9997971
Median	1849.61	0.9997969	1834.06	0.9997961	1882.54	0.9997976
SD	523.55	1.07E-05	627.34	1.13E-05	762.54	1.21E-05
MAPE	22.66%	0.0009%	24.38%	0.0010%	29.66%	0.0011%

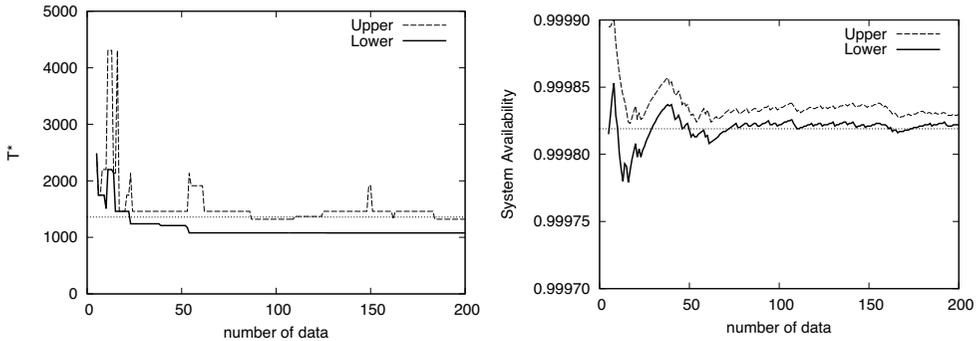


図9 推定値に関する漸近的性質 (Case-II)

繰り返して得られた平均値と中央値に関する漸近的な振る舞いを示す。図10~12より、最適若化スケジュール推定値の平均値と中央値はいずれも n が小さい時には理論的な最適解よりも大きく、 n が大きくなると小さくなる様子が確認できる。一方で最大アベイラビリティの推定値は理論的な最大値を上限と下限で挟み込むような結果になっている。

図13には推定値の標準偏差に関する漸近的な振る舞いを示す。図13より、推定値の標準偏差はデータ数の増加に伴って小さくなる事が確認できる。

図14には推定値の平均絶対誤差率に関する漸近的な振る舞いを示す。この結果は図8や表5, 6に示した結果と大きく異なることに注意を要する。つまり、図14では、 n が小さい時には下限の平均絶対誤差率が上限よりも小さくなっているが、 n が大きくなると逆転し上限の平均絶対誤差率が小さくなっている様子を読み取ることができる。この結果は表7と図15に示す結果を用いて以下のように考察することができる。表7には最適ソフトウェア若化スケジュールの推定値が、新たな障害や打ち切り時間データを観測した後にどの様に変化したかを集計した結果を示している。表7より、上限、下限ともに新たなデータを獲得しても多くの場合で変化しないが、上限の場合は短くなったり長くなったりする割合がほぼ同等に存在する。その一方で、下限の場合は短くなる割合が長くなる割合よりも高くなっている。単一システムで適応的な若化スケジュールを実施した場合、一旦短い予防若化スケジュールが設定されるとその時刻までに障害が発生しなかった場合は予防的若化が実施されるため、それらのデータがすべて打ち切りデータとなってしまう。このとき、下限のスケジューリングを採用した場合には次の若化スケジュールが長くなる事がほとんどなく、変化しないか短くなる傾向があるため、図15に示すとおりますます予防若化によってデータが打ち切れやすくなる。上限についても類似した結果が確認できるものの、若化スケジュールの推定値が短くなる傾向は下限よりも小さく、

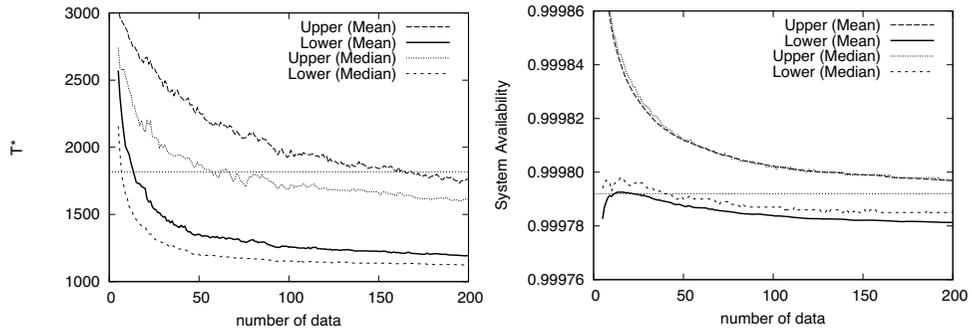


図 10 推定値の平均値および中央値に関する漸近的性質 (Case-I)

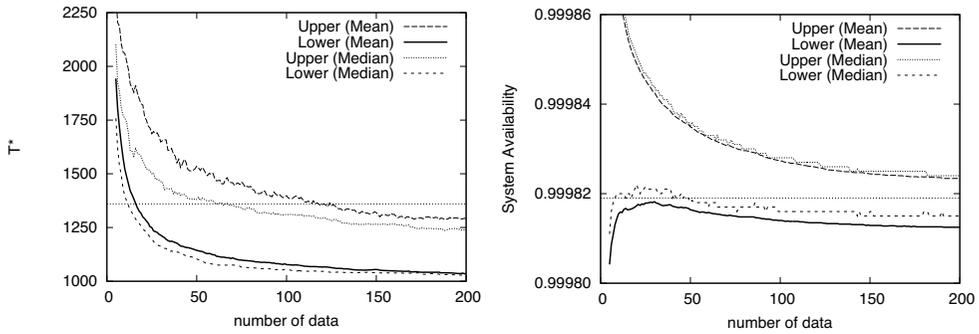


図 11 推定値の平均値および中央値に関する漸近的性質 (Case-II)

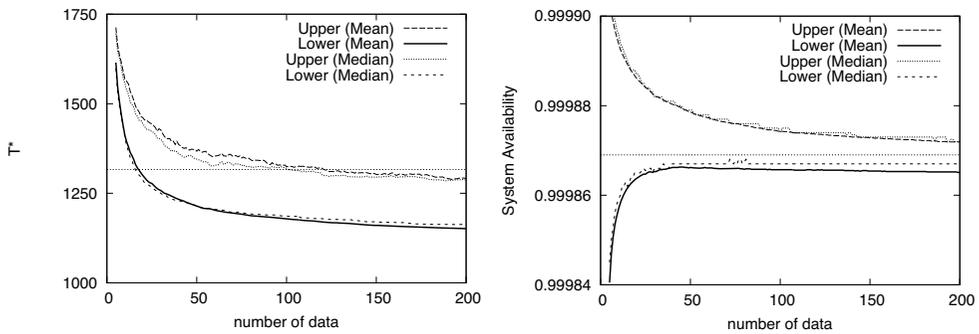


図 12 推定値の平均値および中央値に関する漸近的性質 (Case-III)

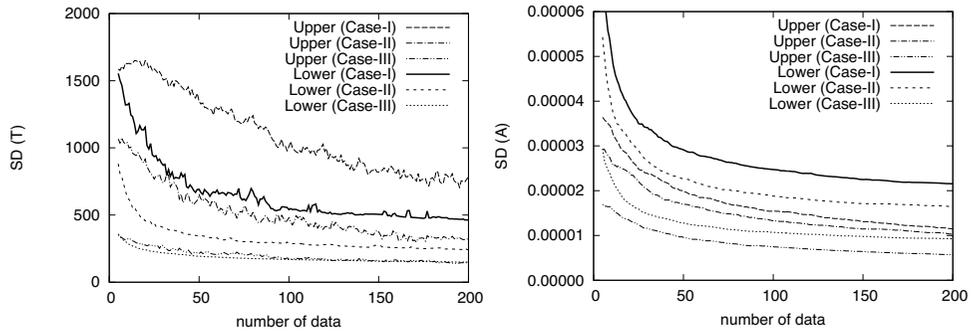


図 13 推定値の標準偏差に関する漸近的性質

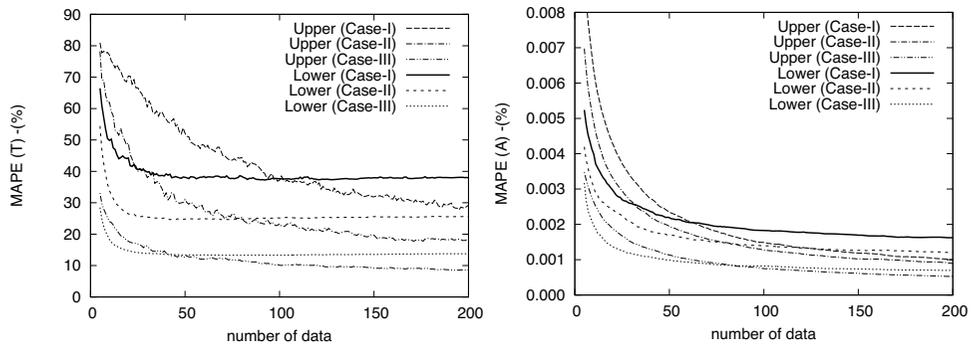


図 14 推定値の平均絶対誤差率に関する漸近的性質

打ち切り確率も下限のそれよりも小さいことがわかる。したがって、 n が大きくなると下限と上限の平均絶対誤差率が逆転することになる。

以上の結果から、ランダム打ち切りデータが得られる状況では、本研究で提案したノンパラメトリック予測推論に基づいたアプローチが有効であると結論づけることができる。また適応的な最適若化スケジューリングについても、提案手法は有効であるものの、大半のデータが予防若化によって打ち切られてしまうような状況になった場合にはあらためて障害時間データが観測できるように環境を整えることも求められる。

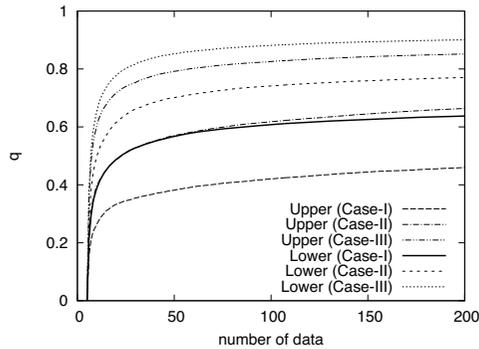


図 15 打ち切り時間データが含まれる割合の推移

表 7 最適ソフトウェア若化スケジュール推定値の変化

	Upper			Lower		
	$T_{up,n+1}^*$ $> T_{up,n+2}^*$	$T_{up,n+1}^*$ $= T_{up,n+2}^*$	$T_{up,n+1}^*$ $< T_{up,n+2}^*$	$T_{low,n+1}^*$ $> T_{low,n+2}^*$	$T_{low,n+1}^*$ $= T_{low,n+2}^*$	$T_{low,n+1}^*$ $< T_{low,n+2}^*$
Case I						
Failure	6.4%	86.7%	6.9%	3.4%	95.4%	1.3%
Censor	6.9%	89.7%	3.4%	2.6%	97.0%	0.4%
Case II						
Failure	6.9%	85.3%	7.8%	3.2%	96.0%	0.8%
Censor	6.1%	90.7%	3.1%	2.4%	97.4%	0.3%
Case III						
Failure	5.1%	89.2%	5.8%	2.8%	96.9%	0.2%
Censor	4.5%	93.0%	2.6%	1.9%	98.0%	0.1%

6. むすび

本論文ではエージング現象に起因するシステム障害が発生するようなソフトウェアシステムに対して、その障害時間データと右側打ち切り時間データが得られる状況を想定し、最適若化スケジュールをノンパラメトリック予測推論によって導出することを考えた。具体的には u 個の障害時間データおよび v 個の右側打ち切り時間データから構成される $n = u + v$ 個のランダム打ち切りデータから、 $n + 1$ 番目の障害発生時間の確率分布と予測信頼度関数の上下限をノンパラ

メトリック予測推論によって定式化し、システムアベイラビリティを最大にするソフトウェア若化スケジュールを適応的に導出した。シミュレーション実験を通じて、ランダム打ち切りデータが逐次得られるような状況では本論文で提案した最適若化スケジュールリングは有効に機能し、新たなデータを逐次得ることでより高精度に最適若化スケジュールとシステムアベイラビリティを推定可能になることが示された。

一方で、単一システムに対する適応的な最適若化スケジュールリングについても提案手法は有効であるものの、提案手法によって得られた最適若化スケジュールを採用し続けた場合に予防若化によって大半のデータが打ち切られてしまうような状況下では、あらためて障害時間データを観測できるような環境構築の必要性も指摘された。今後の課題として、このような障害データを観測するためのしきい値設定の最適化を取り扱ったり、期待費用およびコスト有効性など他の評価規範に対してもノンパラメトリック予測推論アプローチを適用することが考えられる。

参考文献

- [1] Adams, E. N.: Optimizing preventive service of software products, *IBM Journal of Research & Development*, Vol. 28, No. 1, pp. 2–14 (1984).
- [2] Marshall, E.: Fatal error: How Patriot overlooked a Scud, *Science*, Vol. 255, No. 5050, p. 1347 (1992).
- [3] Avritzer, A. and Weyuker, E. J.: Monitoring smoothly degrading systems for increased dependability, *Empirical Software Engineering*, Vol. 2, No. 1, pp. 59–77 (1997).
- [4] Yurcik, W. and Doss, D.: Achieving fault-tolerant software with rejuvenation and reconfiguration, *IEEE Software*, Vol. 18, No. 4, pp. 48–52 (2001).
- [5] Castelli, V., Harper, R. E., Heidelberger, P., Hunter, S. W., Trivedi, K. S., Vaidyanathan, K. and Zeggert, W. P.: Proactive management of software aging, *IBM Journal of Research & Development*, Vol. 45, No. 2, pp. 311–332 (2001).
- [6] Shereshevsky, M., Crowell, J., Cukic, B., Gandikota, V. and Liu, Y.: Software aging and multifractality of memory resources, in *Dependable Systems Networks (DSN)*, *IEEE International Conference on*, pp. 721–730 (2003).
- [7] Grottke, M., Li, L., Vaidyanathan, K. and Trivedi, K. S.: Analysis of software aging in a web server, *Reliability, IEEE Transactions on*, Vol. 55, No. 3, pp. 411–420 (2006).
- [8] Matias Jr, R. and Filho, P. J. F.: An experimental study on software aging and

- rejuvenation in web servers, in *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, Vol. 1, pp. 189–196, IEEE (2006).
- [9] Grottke, M. and Trivedi, K. S.: Fighting bugs: Remove, retry, replicate, and rejuvenate, *IEEE Computer*, Vol. 40, No. 2, pp. 107–109 (2007).
- [10] Grottke, M., Matias Jr, R. and Trivedi, K. S.: The fundamentals of software aging, in *Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on*, pp. 1–6 (2008).
- [11] Macêdo, A., Ferreira, T. B. and Matias Jr, R.: The mechanics of memory-related software aging, in *Software Aging and Rejuvenation (WoSAR), 2010 IEEE Second International Workshop on*, pp. 1–5 (2010).
- [12] Matias Jr, R., Barbetta, P. A., Trivedi, K. S. and Filho, P. J. F.: Accelerated degradation tests applied to software aging experiments, *Reliability, IEEE Transactions on*, Vol. 59, No. 1, pp. 102–114 (2010).
- [13] Machida, F., Nicola, V. F. and Trivedi, K. S.: Job completion time on a virtualized server subject to software aging and rejuvenation, in *Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on*, pp. 44–49 (2011).
- [14] Yoshimura, T., Yamada, H. and Kono, K.: Can Linux be rejuvenated without reboots?, in *Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on*, pp. 50–55 (2011).
- [15] Huang, Y., Kintala, C., Kolettis, N. and Fulton, N. D.: Software rejuvenation: Analysis, module and applications, in *Fault Tolerant Computing (FTC), Twenty-Fifth International Symposium on*, pp. 381–390 (1995).
- [16] Dohi, T., Goševa-Popstojanova, K. and Trivedi, K. S.: Estimating software rejuvenation schedules in high-assurance systems, *Computer Journal*, Vol. 44, No. 6, pp. 473–485 (2001).
- [17] Dohi, T., Iwamoto, K., Okamura, H. and Kaio, N.: Discrete availability models to rejuvenate a telecommunication billing application, *IEICE Transactions on Communications (B)*, Vol. E86-B, No. 10, pp. 2931–2939 (2003).
- [18] Vaidyanathan, K. and Trivedi, K. S.: A comprehensive model for software rejuvenation, *Dependable and Secure Computing, IEEE Transactions on*, Vol. 2, No. 2, pp. 124–137 (2005).
- [19] Rinsaka, K. and Dohi, T.: Optimizing software rejuvenation schedule based on the

- kernel density estimation, *Quality Technology and Quantitative Management*, Vol. 6, No. 1, pp. 55–65 (2009).
- [20] Rinsaka, K. and Dohi, T.: A faster estimation algorithm for periodic preventive rejuvenation schedule maximizing system availability, in *Service Availability*, pp. 94–109, Springer (2007).
- [21] Zhao, J., Wang, Y. B., Ning, G. R., Trivedi, K. S., Matias Jr, R. and Cai, K.-Y.: A comprehensive approach to optimal software rejuvenation, *Performance Evaluation*, Vol. 70, No. 11, pp. 917–933 (2013).
- [22] Coolen, F. P. A., Coolen-Schrijner, P. and Yan, K. J.: Nonparametric predictive inference in reliability, *Reliability Engineering & System Safety*, Vol. 78, No. 2, pp. 185–193 (2002).
- [23] Hill, B. M.: Posterior distribution of percentiles: Bayes’ theorem for sampling from a population, *Journal of the American Statistical Association*, Vol. 63, No. 322, pp. 677–691 (1968).
- [24] Hill, B. M.: Parametric models for A_n : Splitting processes and mixtures, *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 55, No. 2, pp. 423–433 (1993).
- [25] Coolen, F. P. A. and Newby, M. J.: Guidelines for corrective replacement based on low stochastic structure assumptions, *Quality and Reliability Engineering International*, Vol. 13, No. 4, pp. 177–182 (1997).
- [26] Coolen, F. P. A. and Coolen-Schrijner, P.: Condition monitoring: A new perspective, *Journal of the Operational Research Society*, Vol. 51, No. 3, pp. 311–319 (2000).
- [27] Coolen-Schrijner, P. and Coolen, F. P. A.: Non-parametric predictive inference for age replacement with a renewal argument, *Quality and Reliability Engineering International*, Vol. 20, No. 3, pp. 203–215 (2004).
- [28] Coolen-Schrijner, P. and Coolen, F. P. A.: Adaptive age replacement strategies based on nonparametric predictive inference, *Journal of the Operational Research Society*, Vol. 55, No. 12, pp. 1281–1297 (2004).
- [29] Coolen, F. P. A. and Coolen-Maturi, T.: Predictive inference for system reliability after common-cause component failures, *Reliability Engineering & System Safety*, Vol. 135, pp. 27–33 (2015).
- [30] Coolen, F. P. A. and Yan, K. J.: Nonparametric predictive inference with right-

- censored data, *Journal of Statistical Planning and Inference*, Vol. 126, No. 1, pp. 25–54 (2004).
- [31] Kaplan, E. L. and Meier, P.: Nonparametric estimation from incomplete observations, *Journal of the American Statistical Association*, Vol. 53, No. 282, pp. 457–481 (1958).
- [32] Berliner, L. M. and Hill, B. M.: Bayesian nonparametric survival analysis, *Journal of the American Statistical Association*, Vol. 83, No. 403, pp. 772–779 (1988).
- [33] 林坂弘一郎, 土肥正: ソフトウェアの最適若化スケジューリングに対するノンパラメトリック予測推論: 適応的スケジューリング, 神戸学院大学経営学論集, Vol. 4, No. 1, pp. 31–53 (2007).
- [34] 林坂弘一郎, 土肥正: ノンパラメトリック予測推論に基づいたソフトウェアの適応的3ステップ若化方策, 神戸学院大学経営学論集, Vol. 12, No. 2, pp. 37–66 (2016).
- [35] Rinsaka, K. and Dohi, T.: Non-parametric predictive inference of preventive rejuvenation schedule in operational software systems, in *Software Reliability Engineering, 2007. ISSRE'07. The 18th IEEE International Symposium on*, pp. 247–256 (2007).
- [36] Rinsaka, K. and Dohi, T.: Non-parametric predictive inference of adaptive software rejuvenation schedule, in *Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on*, pp. 1–6 (2008).
- [37] Rinsaka, K. and Dohi, T.: Toward high assurance software systems with adaptive fault management, *Software Quality Journal*, Vol. 24, No. 1, pp. 65–85 (2016).
- [38] Leemis, L. M.: *Reliability: Probabilistic Models and Statistical Methods*, Prentice-Hall, Inc. (1995).