マイクロコンピュータとワークステーション 64－5
（1990．11．30）

フォールトトレラント・システムの信頼性評価シミュレータのワークステーション上での実現

土井強 ， 向殿政男
明治大学 理工学部

フォールトトレラント・システムを評価する方法の一つに，離散型マルコフ連鎖変換法があり，これに基づいた信頼性尺度評価プログラムＲＥＳＯＲＴが開発されている．これは従来大型計算機上で実現されていたが，今回ＷＳ上に移植を行なった．これを用いてシステムの評価を行なう場合，システムをマルコフモデルによって表す必要がある．そのため，マルコフモデルを自動生成するプログラムが開発されていたが，やや複雑なシステムでは，単純にモデルを生成すると規模が大きくなり過ぎシミュレーションの実行が困難となる場合が生じていた．そこで，今回は簡単化されたモデルを生成するプログラムを開発し，従来の方法との比較を行なった．

Inplementation of Simulator to evaluate Fault Tolerant System's reliability on EWS

Tsuyoshi Doi , Masao Mukaidono
School of Science and Technology, Meiji University

The ramdomization is one of the techniques to evaluate Fault Tolerant Systems. Based on this technique, an availavility evaluation program RESORT was developed. This has been carried out only on mainframe computers. In this paper, we implement it on EWS. For evaluating Fault Tolerant Systems, we need to represent it by Marcov model. The program that generates Marcov model automatically has been developed. But when we apply it to complicated system, we have some difficulties to carry out the simulator because generated model is too large. So we developed a program to generate simplified model and compared it with former method.

1 はじめに

フォールトトレラント・システムを評価する方法の１つに，離散型マルコフ連鎖変換法（Randomization Tequniques）があり，これに基づいた信頼性尺度評価プログラムＲＥＳＯＲＴ（Reliavility Evaluation SOfutware by Randomization Tequniques）が開発されている[1]．ＲＥＳＯＲＴは実行時に比較的大きな記憶容量を必要とすることなどから，従来大型計算機上で実現されていたが，今回ワークステーション上に移植を行なった．これを用いてシステムの信頼性評価を行なう場合，システムをマルコフモデルによって表現する必要がある．そのため，マルコフモデルを自動生成するプログラムが開発されていた[2]が，やや複雑なシステムでは，単純にモデルを生成すると規模が大きくなり過ぎシミュレーションの実行が困難となるなどの問題点があった．そこで，簡単化されたモデルを生成するプログラムを開発し，従来の方法との比較実験を行なった．

2 離散型マルコフ連鎖変換法

フォールトトレラント・システムを，確率モデルによって評価する場合，定常解と過渡解の２つが考えられる．定常解は，時刻ｔが無限大の定常状態におけるシステムの挙動を示すものである．これは比較的簡単な手法を用いて，解析的に求めることができる．一方，過渡解は，ある時刻ｔにおけるシステムの挙動を示すもので，その解は時刻ｔの関数となる．これは一部の場合を除いて，解析的に求めることは困難である．

システムの故障率及び修理率の確率分布が全て指数分布であると仮定すれば，そのシステムの状態を表す確率モデルはマルコフ過程となる．ここでマルコフ過程とは，対象とするモデルの状態が現時点の状態だけから定まり，過去の状態に影響されないような過程を言う．マルコフ過程の過渡解は，離散型マルコフ連鎖変換法を用いて，数値的に求めることができる．

いま，マルコフ過程の有限個の状態を$i=1, 2, \cdots, N$とすれば，時刻t（≥ 0）で状態iである確率を$\pi i(i)$として，Ｎ要素の行ベクトルを

$$\Pi(t)=[\pi 1(t), \pi 2(t), \cdots, \pi N(t)] \qquad (1)$$

とすれば，状態確率ベクトル$\Pi(t)$は，微分方程式

$$\frac{d\Pi(t)}{dt}=\Pi(t)Q \qquad (2)$$

の解として

$$\Pi(t)=\Pi(0)e^{Qt} \qquad (3)$$

により計算できる．但し，$\Pi(0)$は初期確率ベクトルである．また行列Ｑはマルコフ過程の無

限小生成行列と呼ばれ，Qの非対角要素を$q_{ij} \geqq 0$（$i \neq j$）としたとき，各対角要素は，

$$-q_i = -\sum_{\substack{j=1 \\ j \neq i}}^{N} q_{ij} \qquad (i = 1,\ 2,\ \cdots,\ N)$$

で与えられる．

次に，行列Pを以下のように定義する．

$$P = Q / \Lambda + I, \qquad \Lambda = \max[q_i] \qquad (4)$$

式（4）で得られる行列Pは確率推移行列となる．つまりPのある要素p_{ij}は，状態 i から状態 j へ推移する確率を表している．式（3），（4）より，

$$\begin{aligned} \Pi(t) &= \Pi(0)\, e^{\Lambda(P-I)t} \\ &= \Pi(0)\, e^{\Lambda t P} \cdot e^{-\Lambda t I} \\ &= \Pi(0) \sum_{n=0}^{\infty} \frac{(\Lambda t)^n}{n!} P^n \cdot e^{-\Lambda t} I \\ &= \sum_{n=0}^{\infty} \frac{(\Lambda t)^n}{n!} e^{-\Lambda t} \cdot \Phi(n) \qquad (5) \end{aligned}$$

但し，

$$\Phi(n+1) = \Phi(n)\, P, \qquad \Phi(0) = \Pi(0)$$

である．このとき，$\Phi(n) = [\phi_1(n),\ \phi_2(n),\ \cdots,\ \phi_N(n)]$ について考えると，行列Pが確率推移行列であることから，$\Phi(n)$は行列Pによるマルコフ連鎖の第nステップの状態確率ベクトルである．また，係数である$(\Lambda t)^n / n!$ は，ポアソン過程の確率分布である．従って，マルコフ過程の状態確率は，ポアソン過程とマルコフ連鎖の積の無限級数で表わされる．

3　シミュレーションプログラム：RESORT

RESORTは離散型マルコフ連鎖変換法を実行するプログラムである．実行に必要なデータは以下に示すものである．

1）マルコフモデルの状態数　N
2）無限小生成行列　Q
3）初期確率ベクトル　$\Pi(0)$
4）計算させたい時刻数　M
5）計算させたい時刻列　（$t_1,\ t_2,\ \cdots,\ t_m$）
6）ポアソン分布項の受容誤差　ε_1
7）マルコフ連鎖項の受容誤差　ε_2

上記１）～６）の値を式（５）に代入すれば，マルコフ過程の状態確率Π（ｔ）を求めることができる．しかし，式（５）は無限級数であるから，そのままでは直接計算機で解くことはできない．そこで，ポアソン分布項が１に収束することに着目し，Π（ｔ）を求める際，十分小さい正数$\varepsilon 1$に対して，

$$T(\varepsilon 1,\ t) = \min \left[k : \sum_{n=0}^{k} \frac{(\Lambda t)^n}{n!} e^{-\Lambda t} > 1 - \varepsilon 1 \right] \qquad (6)$$

なる打切り回数Ｔ（ε1，ｔ） を設定している．また，このマルコフ連鎖が吸収状態を持たず，さらに非周期的であるならば，マルコフ連鎖の状態確率Φ（ｎ）は定常状態確率に収束する．そこで，十分小さい正数$\varepsilon 2$に対して，

$$N(\varepsilon 2) = \min \left[n : |\Phi(n+1) - \Phi(n)| < \varepsilon 2 \right] \qquad (7)$$

なるΦ（Ｎ（ε2））を定常状態確率としても問題は無いと考えられる．この収束受容誤差の概念を導入することにより，実行時に必要な記憶容量の削減，及び実行時間の短縮が期待できる．

このようにして求められた状態確率Π（ｔ）より，各要素に重み係数ｗ（ｉ）（ｉ＝１～状態数）を乗じてその総和を取ることで，いろいろな信頼性尺度を得ることができる．以下にその代表例を示す．

（１）アベイラビリティ

$$w(i) = \begin{cases} 1 & \text{Operating} \\ 0 & \text{Failed} \end{cases}$$

（２）計算アベイラビリティ

$$w(i) = \begin{cases} \alpha i & \text{Operating} \quad (\alpha i\text{：計算容量}) \\ 0 & \text{Failed} \end{cases}$$

なお，ＲＥＳＯＲＴはこれまで大型計算機上でＦＯＲＴＲＡＮによって実現されていたが，今回ワークステーション上へ移植するにあたり，Ｃ言語に変更し，合わせて実行時間の短縮と実行時に必要な記憶容量の削減を図った．これにより，小規模なモデルであれば，従来不可能であったパーソナルコンピュータ上でもシミュレーションの実行が可能となった．

４　マルコフモデルの自動生成

４．１　従来の自動生成プログラム

ＲＥＳＯＲＴで必要なマルコフモデルとそのＱ行列の作成は，比較的単純な構成のシステムであっても，その状態数が大きなものとなるため，非常に面倒であり，データを作成する際に人為的ミスを起こしやすい．そこで，マルコフモデルとＱ行列を自動的に生成するプログラムが開発

されている[2]．例として図１のような２種類のユニットからなるシステムを考える．このシステムのマルコフモデルは図２のようになる．図中の（）内は，その状態で故障しているユニットを表している．なお，システムダウン後の故障は起こらないものと仮定している．

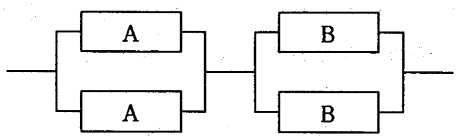

図１．フォールトトレラントなシステム構成の例

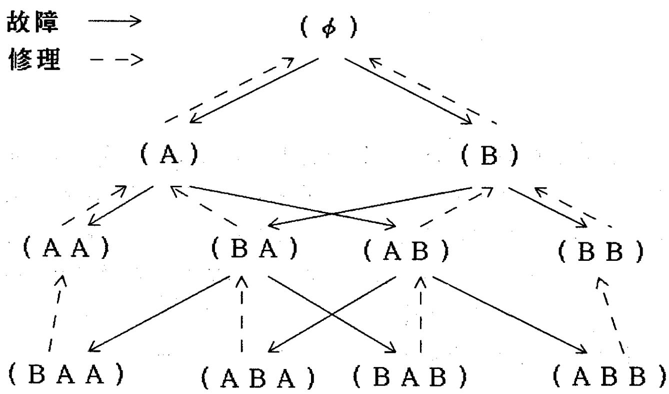

図２．マルコフモデルの状態遷移図

図１の例の場合，システムの構成として，各ユニットの故障率，修理率，接続状態（現在は論理式表現を用いている）を表１のような形式で入力すれば，マルコフモデルとＲＥＳＯＲＴで使用する書式のＱ行列がファイルとして出力される．

2			：	ユニットの個数
A	０．００１	０．２	：	ユニットの名前，故障率，修理率
B	０．００３	０．４		
（A＋A）＊（B＋B）			：	ユニットの接続状態

表１．自動生成プログラムの入力データ例

４．２　新しい修理方策

前節で述べた従来の修理方策は，図２のように，単純に先に故障したユニットから順に修理をしていくというものである．しかし，この方策では，システムダウンの状態であるときに，１つのユニットを修理してもやはりシステムダウンの状態へ移ってしまうことが考えられる．この場

合，別のユニットを修理すればシステムは稼動状態となるので，この方策は現実的ではなく，またアベイラビリティの点からも不利である．つまり，図中で

（ＡＢＢ）のシステムダウン状態から，最初に故障したＡのユニットを修理すると（ＢＢ）の状態へ移るが，これもまたシステムダウン状態である．このとき，Ｂのユニットを修理すれば，稼働状態である（ＡＢ）へ移る．

そこで，ユニットに優先順位をつけて，あるユニットを修理したとき，その結果がシステムダウン状態とならない，最も優先度の高いユニットを修理するという方策を考える．優先順位のつけ方として，

① 先に故障した順
② 故障率の高い順
③ 故障率の低い順
④ 修理率の高い順
⑤ 修理率の低い順

の５通りを用いる．図３に，表１の例に各修理方策を適用して生成した各々のマルコフモデルを用いて，ＲＥＳＯＲＴでアベイラビリティを求めた結果を示す．この例の場合，②と④，及び③と⑤の方策で生成されたモデルはそれぞれ等しくなる．図より明らかに，新しい修理方策の方がアベイラビリティが高くなっていることがわかる．なお，この新しい修理方策の概念は，前節で述べた自動生成プログラムにおいても多少考慮されていたが，今回ワークステーション上に移植するにあたって，次節で述べる状態数の簡単化とあわせて全面的に取り入れた．

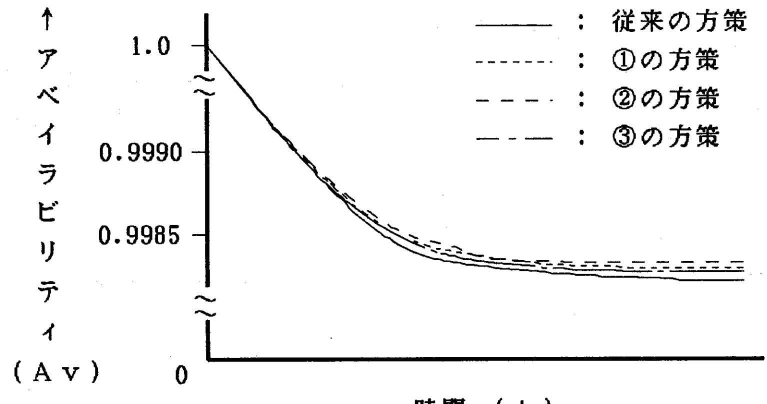

図３．修理方策によるアベイラビリティの相違

４．３ 状態数の簡単化

図２の例では，同じユニットが同じ数だけ故障していて，故障した順序だけが異なっている，複数の状態を含んでいる．しかし，前節で述べた新しい修理方策のうち，ユニットの修理される優先順位が，ユニットの故障した順番に依らないもの（即ち①以外）では，このような状態は互

いに区別する必要がないと考えられる．そこで図4のようにマルコフモデルの簡単化を行っても，アベイラビリティに関しては等価である．ＲＥＳＯＲＴでシミュレーションを実行する場合，状態数は1つでも少ない方が，実行に要する記憶容量，及び実行時間の点で有利である．

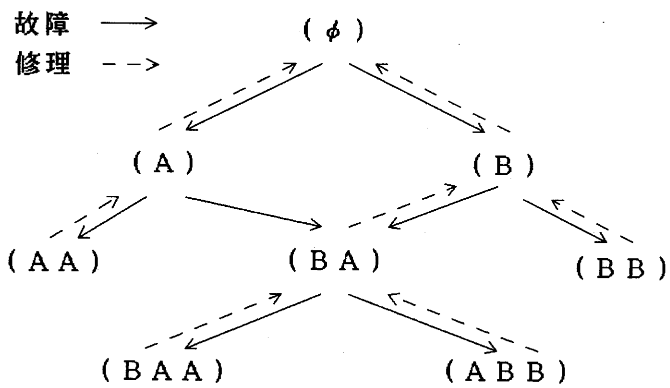

図4．簡単化されたマルコフモデルの例

この例では，状態数が１１から８に減少しているに過ぎない．しかし，

$$A*B*((C+D+E+F+G)*(C+D+E+F+G))*H*I$$

なる論理式で表される構成のシステムでは，従来の方策で生成されたマルコフモデルの状態数が２９３５であったのに対し，簡単化されたモデルでは２４０と大幅に減少している．これらのことから，状態数の簡単化を行うことは極めて有効であるといえる．

5．まとめ

シミュレーションプログラムＲＥＳＯＲＴ，及びその入力データであるマルコフモデルを生成するプログラムは，既に大型計算機上で実現されていたが，双方ともあまり大きなモデルを取り扱うことができないという欠点があった．今回，これらをワークステーション上に移植し，またマルコフモデルを生成する際の新しい方策を実現するとともに，その状態数を簡単化する方法を提案した．

参考文献

［１］日本規格協会：”システムの高信頼性技術に関する調査研究
（電子応用システム）報告書”，昭和６０年度

［２］福嶋：”マルコフモデルの自動生成とフォールトトレラントシステムの評価への応用”，
明治大学修士論文，(1989,1)