

論文

タグ付きタスクモデルの一解法について

今井 博英[†] 角山 正博^{††} 石井 郁夫[†] 牧野 秀夫^{†††}

An Analyzing Method for Tagged-Task-Models

Hiroei IMAI[†], Masahiro TSUNOYAMA^{††}, Ikuo ISHII[†], and Hideo MAKINO^{†††}

あらまし 少数の吸収状態と多数の過渡状態からなるタグ付きタスクモデルの状態確率を、そのモデルの特徴を生かして少ない計算量で求める方法を示す。ここでは、タグ付きタスクモデルの状態を到達可能性に基づく同値類に分割し、その同値類を更に再帰的に分割することによって計算に必要な記憶容量を減らす方法を示す。更に、状態確率の小さくなった同値類を集約し、一つの状態で表すことによって計算時間を減らす方法を示す。最後に、実際に数値計算を行って得られた計算量および誤差の例を示す。

キーワード タグ付きタスクモデル, 連続時間型マルコフ連鎖, リアルタイムシステム, 性能解析

1. まえがき

リアルタイムシステムが大規模化し適用分野が広がるのに伴い、システムの性能の解析および評価、特にシステム中に故障が発生した場合やシステムへの入力が急激に変化した場合等の過渡的な状態における性能の解析および評価が重要になってきている。このようなリアルタイムシステムの解析方法としてはシミュレーションによる方法の他に、ペトリネット [1],[2] や待ち行列 [4] を用いる方法等が提案されている。しかし、高精度な解析を行うためには、シミュレーションによる方法では非常に多くの計算を必要とするため時間的および経済的に問題があること、また後者の方法はいずれも定常状態における解析を行うものが大部分であるため、過渡状態における解析には不向きであること等の問題点があった。この課題に対処し、過渡状態における解析を効率良く行うためのシステムのモデル化の一つの方法として、タグ付きタスクモデルに基づくシステムの解析方法が提案されている [2],[3]。これは、システム中のある一つのタスクに注目し、そのタ

スクの処理が終了するまでの過程をモデル化することによって、システムの性能を解析する方法である。筆者らはこの方法を用いて、タグ付きタスクの処理過程を連続時間型マルコフ連鎖を用いてモデル化することにより、リスポンシブシステムにおけるプロセッサ故障時の過渡状態の性能を解析する方法を提案した [5]。

しかし、このようなモデル化を行った場合には、システムが大規模化し、あるいは処理内容が複雑化するに伴って、状態数が急速に増大するために解析が困難になるという問題があった。このような大規模なマルコフ連鎖を解析する方法として、文献 [3] では、初期状態において状態確率が小さい状態を切り捨て、残りの状態のみを用いて計算を行う方法が提案されている。しかし、モデルの構造的な特徴が考慮されていない。また、文献 [7],[8] では、既約なマルコフ連鎖の定常状態の状態確率を集約と分割を繰り返すことによって求める方法が述べられており、文献 [9] ではマルコフ連鎖の状態を推移率の値が異なる大小 2 種類の集合に分割し、推移率の大きい状態同士を集約することによって解析する方法が提案されている。しかし、筆者らが提案したタグ付きタスクモデルは既約ではなく、また推移率の値にも大きな差がないためこれらの方法を適用することができないという問題がある。

本論文では、タグ付きタスクモデルの構造的な特徴を考慮し、互いに到達可能な状態を同値類としてまとめて集約しながら計算をすることによって、計算に必要な記憶容量が少なくかつ短時間に解析を行うことの

[†] 新潟大学大学院自然科学研究科, 新潟市
Graduate School of Science and Technology, Niigata University,
8050, Igarashi-2, Niigata-shi, 950-21 Japan

^{††} 新潟工科大学, 柏崎市
Niigata Institute of Technology, 1719, Kashiwazaki-shi, 945-11
Japan

^{†††} 新潟大学工学部, 新潟市
Faculty of Engineering, Niigata University, 8050, Igarashi-2,
Niigata-shi, 950-21 Japan

できる方法を示す。まず 2. で解析の対象となるリアルタイムシステムおよびタグ付きタスクモデルの概要を示す。続いて 3. ではタグ付きタスクモデルの性質を考慮することによって、状態確率を求めるために必要な記憶容量を削減する方法を示し、4. では計算時間を削減する方法を示すと共に、誤差の上限を示す。5. では実際のシステムの解析例を示し、最後に 6. でまとめを行う。

2. 諸定義

2.1 リアルタイムシステム

本論文で検討を行うリアルタイムシステム [5] は s 個のプロセッサと一つの待ち行列からなっている。ここでプロセッサ数と待ち行列の容量の和 L をシステムの容量と呼ぶ。

このシステムで処理されるタスクは、1 から M までの M 個のクラスに分類されている。更にクラス j ($1 \leq j \leq M$) のタスクは、同じ発生間隔、処理時間および制限時間分布に従っている。なお、これらの確率分布はいずれもパラメータ λ_j, μ_j, ν_j の指数分布とする。なお、これらの確率分布が指数分布でないシステムについても、与えられた確率分布を Γ 分布で近似し、更にパラメータに制限を加えることによって、これを指数分布に分解できるため [11]、本手法を適用することができる。

またタスクのクラスの番号が小さいものほど優先度が高いものとし、異なる優先度のタスクはプリエンパティプに処理され、同一優先度のタスクは先着順に処理されるものとする。

2.2 タグ付きタスクモデル

クラス j に属する代表的な一つのタスクに注目し、その処理の過程をモデル化した連続時間型マルコフ連鎖 $\{X(t)\}_{t \geq 0}$ をタグ付きタスクモデルと呼び [5]、この状態集合を S で表す。タグ付きタスクモデルの状態は、タグ付きタスクの処理が終了したことを表す状態 C と、制限時間を超えたためあるいは待ち行列があふれたために捨てられたことを表す状態 F 、更にタグ付きタスクがシステム中にあり、タグ付きタスクよりも先に処理されなければならないクラス k ($1 \leq k \leq j$) のタスクの個数が i_k 個あることを表す状態 $(i_1, \dots, i_k, \dots, i_j)$ からなる。状態 C と F は一度その状態に推移した後では他の状態へ推移することはないため、吸収状態となる。

状態集合 S を吸収状態と過渡状態に分類し、それ

ぞれ $S_A = \{C, F\}$, $S_T = \{(i_1, \dots, i_k, \dots, i_j) | \sum_{k=1}^j i_k < L\}$ で表す。状態 α から β への推移率を $g_{\alpha\beta}$ で表し、推移率行列を $G = [g_{\alpha\beta}]_{\alpha, \beta \in S}$ で表す。状態の遷移はタスクの発生、処理終了および制限時間を超えたために捨てられることによって起こる。これらの推移率はそれぞれ λ_j, μ_j, ν_j で表されるため、 $g_{\alpha\beta}$ はこれらの推移率の和で表される [5]。

次に時刻 t における状態が α である確率を $p_\alpha(t)$ で表し、状態確率を要素とする横ベクトル $p(t) = [p_\alpha(t)]_{\alpha \in S}$ を状態確率ベクトルと呼ぶ。特に、 $t = 0$ での状態確率ベクトルを初期状態確率ベクトルと呼ぶ。

状態確率および推移率行列を吸収状態 S_A と過渡状態 S_T の二つの部分集合に関して次のように構成し直す。

$$G = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ G_{TA} & G_{TT} \end{pmatrix} \quad (1)$$

$$p(t) = (p_A(t), p_T(t)) \quad (2)$$

時刻 t におけるマルコフ連鎖の状態確率は、初期状態確率 $p(0)$ を初期条件として次式を解くことによって求めることができる [10]。

$$\frac{dp(t)}{dt} = p(t)G \quad (3)$$

この微分方程式を刻み幅 Δt で離散化すると次式が得られる。

$$\begin{aligned} p(t + \Delta t) &= p(t)(I + G\Delta t) \\ &= p(t)Q \end{aligned} \quad (4)$$

ここで、 $Q = I + G\Delta t$ であり、 I は単位行列を表す。また Q を推移確率行列と呼ぶ。 $Q = [q_{\alpha\beta}]$ についても G と同様に、 S_A と S_T に関して構成し直すことによって次式が得られる。

$$Q = \begin{pmatrix} I & \mathbf{0} \\ Q_{TA} & Q_{TT} \end{pmatrix} \quad (5)$$

3. 状態集合の分割による記憶領域の削減

3.1 到達可能性による分割

前章で求めた推移確率行列 Q を用いると、時刻 t における状態確率ベクトルから時刻 $t + \Delta t$ における状態確率ベクトルが次式のように得られる。

$$p_A(t + \Delta t) = p_A(t) + p_T(t)Q_{TA} \quad (6)$$

$$p_T(t + \Delta t) = p_T(t)Q_{TT} \quad (7)$$

タグ付きタスクモデルでは S_T の状態は j 個の系内数の組合せであるため、状態数のオーダーは $O(L^j)$ である。このため、システムの容量やクラス数の増加に伴って状態確率を求めることが困難になる。この問題を解決するために、本論文ではタグ付きタスクモデルが有する状態間の遷移の規則性に着目した計算量の削減を行う。

まず状態間の相互到達可能性を次のように定義する [10].

[定義 1] 状態の相互到達可能性

状態 $\alpha, \beta \in S (\alpha \neq \beta)$ について、1 回以上の推移を繰り返すことによって α から β に推移可能であり、かつ β から α へも推移可能であるとき、状態 α と β は相互到達可能であるといい、 $\alpha \leftrightarrow \beta$ と表す。□

この関係は同値関係であるため、状態集合 S を状態の相互到達可能性に基づいて同値類に分割することができる。次に状態集合の部分集合の間の到達可能性について次のように定義する。

[定義 2] 状態集合の到達可能性

状態 $\alpha \in S_a$ から $\beta \in S_b (a \neq b)$ へ 1 回の推移で到達可能であるような状態 α と β が少なくとも 1 組存在するとき、状態集合 S_a は状態集合 S_b に到達可能であると言い、 $S_a \rightarrow S_b$ で表す。□

過渡状態集合 S_T を状態の相互到達可能性に基づいて同値類に分割し、得られた部分集合をそれぞれ、 S_0, S_1, \dots, S_N で表す。このとき状態の到達可能性に基づいて部分集合の間で $S_a \rightarrow S_b$ ならば $a > b$ となるように番号付けを行うことができる。

クラス j のタスクを解析するためのタグ付きタスクモデルにおいては、同じクラスのタスクは先着順に処理されるので、 i_j は増加することがない。このため、 i_j が異なる状態は互いに到達可能とはならない。一方、 i_1, \dots, i_{j-1} は、優先度の高いタスクがプリエンティブに処理されるため、増加する場合および減少する場合の両方の場合があり得る。このため、 i_j が等しい状態の集合は、互いに到達可能な同値類をなす。このような部分集合 S_a は次のように表すことができる。

$$S_a = \{(i_1, \dots, i_k, \dots, i_{j-1}, a) \mid \sum_{k=1}^{j-1} i_k + a < L\} \\ (a = 0, 1, \dots, L-1) \quad (8)$$

このように分割することにより、同値類の要素の数は

$O(L^{j-1})$ となる。

更に、部分集合 S_a に到達可能な集合は S_{a+1} のみであることから、過渡状態の推移確率行列 Q_{TT} は部分集合に関して次のように構成し直すことができる。

$$Q_{TT} = \begin{pmatrix} Q_{00} & 0 & \cdots & \cdots & 0 \\ Q_{10} & Q_{11} & 0 & \cdots & 0 \\ 0 & Q_{21} & Q_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & Q_{N-1,N} & Q_{N,N} \end{pmatrix} \quad (9)$$

3.2 同値類の再分割

式 (9) で表される行列の各ブロック $Q_{ab} (a, b = 0, 1, \dots, N)$ を格納するために必要な記憶容量は、同値類の要素の数が $O(L^{j-1})$ であることから、 $L^{2(j-1)}$ に比例する。この容量を更に少なくするために次に示すような手順により同値類を再分割する。

なお、状態数に関するしきい値 N_{th} は、使用可能な記憶容量に基づいてあらかじめ与えられているものとする。

以下に示す手順において、クラス h から j の系内数が等しい状態をまとめた集合を $S_{(a_h, \dots, a_j)}$ で表す。ここで同値類 S_{a_j} と $S_{(a_j)}$ は同じ状態集合を表す。

[状態集合の分割手順]

(1) $h \leftarrow j$ とする。

(2) $S_{(a_h, \dots, a_j)}$ に含まれる状態をクラス $h-1$ の系内数 i_{h-1} の値が等しいものを集めた部分集合に分割する。

(3) 分割した各部分集合の要素数がしきい値 N_{th} よりも大きい場合には h を $h-1$ としてステップ (2) に戻る。そうでないなら手続きを終了する。

上記の手順 (2) を行うことにより、同値類に含まれる状態間の推移率行列 Q_{ab} を更にブロックに分割し、その中ですべての要素が 0 であるブロックの計算を省略することによって、精度を低下させることなく計算に必要な記憶容量と計算時間を削減することができる。また、このようなすべての要素が 0 であるブロックをできるだけ大きくするには、一つの部分集合から推移可能な集合の個数をできるだけ少なくすることが必要である。このために系内数が等しい状態同士をまとめて部分集合を構成することにより、到達可能な部分集合の個数の少ない分割を実現する。

この手順により, $S_{(a_h, \dots, a_j)}$ は次式のような $N_{(a_h, \dots, a_j)} + 1$ 個の部分集合に分割される.

$$S_{(a_{h-1}, a_h, \dots, a_j)} = \{(i_1, \dots, i_k, \dots, i_{h-2}, a_{h-1}, a_h, \dots, a_l, \dots, a_j) \mid \sum_{k=1}^{h-2} i_k + \sum_{l=h-1}^j a_l < L\} \quad (a_{h-1} = 0, \dots, N_{(a_h, \dots, a_j)}) \quad (10)$$

ここで, $S_{(a_h, \dots, a_j)}$ を分割してできる部分集合の数 $N_{(a_h, \dots, a_j)}$ は a_{h-1} のとり得る値の数で決定される. 系内数の和は L 未満であり, クラス h から j の系内数の和が $\sum_{l=h}^j a_l$ であるので, $N_{(a_h, \dots, a_j)}$ は次式で求められる.

$$N_{(a_h, \dots, a_j)} = L - \sum_{l=h}^j a_l \quad (11)$$

このようにして分割された部分集合 $S_{(a_h, \dots, a_j)}$ の状態数は, クラス 1 からクラス $h-1$ までの系内数の組合せで決まるため, 最大の部分集合の状態数のオーダーは $O(L^{h-1})$ である.

状態集合 $S_{(a_h, \dots, a_j)}, S_{(b_k, \dots, b_j)}$ について, $S_{(a_h, \dots, a_j)}$ の状態確率ベクトル, および, $S_{(a_h, \dots, a_j)}$ から $S_{(b_k, \dots, b_j)}$ への推移確率行列を次のように表す.

$$\mathbf{p}_{(a_h, \dots, a_j)}(t) = [p_\alpha(t)]_{\alpha \in S_{(a_h, \dots, a_j)}} \quad (12)$$

$$\mathbf{Q}_{(a_h, \dots, a_j)(b_k, \dots, b_j)} = [q_{\alpha\beta}]_{\substack{\alpha \in S_{(a_h, \dots, a_j)} \\ \beta \in S_{(b_k, \dots, b_j)}}} \quad (13)$$

3.3 状態確率の計算

同値類 $S_{(a_j)}$ に含まれる状態の状態確率ベクトルを $\mathbf{p}_{(a_j)}(t)$ で表すと, 式 (7) および式 (9) より, $\mathbf{p}_{(a_j)}(t + \Delta t)$ は次式のように表すことができる.

$$\mathbf{p}_{(a_j)}(t + \Delta t) = \begin{cases} \mathbf{p}_{(a_j)}(t)\mathbf{Q}_{(a_j)(a_j)} + \mathbf{p}_{(a_j+1)}(t)\mathbf{Q}_{(a_j+1)(a_j)} & (a_j = 0, \dots, N-1) \\ \mathbf{p}_{(a_j)}(t)\mathbf{Q}_{(a_j)(a_j)} & (a_j = N) \end{cases} \quad (14)$$

上式と同値類 $S_{(b_j)}, (b_j = a_j, a_j + 1)$ から $S_{(a_j)}$ への Δt 時間での推移を表す $\mathbf{p}_{(b_j)}(t)\mathbf{Q}_{(b_j)(a_j)}$ を, 前節で分割した部分集合ごとに, 以下の手順に従って計算する.

[$\mathbf{p}_{(b_j)}(t)\mathbf{Q}_{(b_j)(a_j)}$ の計算方法]

(1) $h \leftarrow j, k \leftarrow j$ とする.

(2) 状態集合 $S_{(a_h, \dots, a_j)}, S_{(b_k, \dots, b_j)}$ が再分割されているかされていないかによって場合分けし, それぞれ次のように計算する.

(a) $S_{(a_h, \dots, a_j)}, S_{(b_k, \dots, b_j)}$ がともに分割されていない場合.

分割された推移確率行列を用いてそのまま計算を行う.

(b) $S_{(b_k, \dots, b_j)}$ が分割されている場合

$$\begin{aligned} & \mathbf{p}_{(b_k, \dots, b_j)}(t)\mathbf{Q}_{(b_k, \dots, b_j)(a_h, \dots, a_j)} \\ &= \sum_{b_{k-1}=0}^{N_{(b_k, \dots, b_j)}} \mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(a_h, \dots, a_j)} \end{aligned} \quad (15)$$

上式の $\mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(a_h, \dots, a_j)}, (b_{k-1} = 0, \dots, N_{(b_k, \dots, b_j)})$ は, $k \leftarrow k-1$ としてそれぞれ (2) に戻り計算する.

(c) $S_{(a_h, \dots, a_j)}$ と $S_{(b_k, \dots, b_j)}$ がともに分割されている場合.

$$\begin{aligned} & \mathbf{p}_{(b_k, \dots, b_j)}(t)\mathbf{Q}_{(b_k, \dots, b_j)(a_h, \dots, a_j)} \\ &= \sum_{b_{k-1}=0}^{N_{(b_k, \dots, b_j)}} (\mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(0, a_h, \dots, a_j)}, \\ & \quad \dots, \\ & \quad \mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(a_{h-1}, a_h, \dots, a_j)}, \\ & \quad \dots, \\ & \quad \mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(N_{(a_h, \dots, a_j)}, a_h, \dots, a_j)}) \end{aligned} \quad (16)$$

上式の $\mathbf{p}_{(b_{k-1}, \dots, b_j)}(t)\mathbf{Q}_{(b_{k-1}, \dots, b_j)(a_{h-1}, a_h, \dots, a_j)}, (a_{h-1} = 0, \dots, N_{(a_h, \dots, a_j)}; b_{k-1} = 0, \dots, N_{(b_k, \dots, b_j)})$ は, $h \leftarrow h-1, k \leftarrow k-1$ としてそれぞれ (2) に戻り計算する.

4. 状態の集約による計算時間の削減と誤差

4.1 計算時間の削減方法

過渡状態の状態確率は, 時間の経過と共に単調に減少するため, この性質に基づいて計算時間を短縮することができる. このためには, 状態確率の和が小さい同値類の状態確率を 0 とみなす方法と, 同値類に属す

る状態確率の和を1個の代表状態に集約する方法がある。集約を行って計算を進めた場合には、集約した後も計算が継続されるため、0とみなした場合より誤差が少なくすみ、最悪の場合でも両者の誤差は等しくなる。そこで、ここでは同値類に含まれる状態の確率の総和があらかじめ定められた値より小さくなったときに、同値類に含まれる状態を1個の状態に集約して計算を続行する方法を示す。この方法では、時間の経過と共に計算に用いられる状態数が減少し、1ステップでの計算量を削減することができる。なお、計算の終了条件を決めるパラメータ δ および状態の分割を決めるしきい値 ε は精度を考慮してあらかじめ決定されているものとする。

なお、状態集約を行いながら求めた場合に、時刻 t における状態が α である確率を $p'_\alpha(t)$ で表し、状態確率分布を $\mathbf{p}'(t) = [p'_\alpha(t)]_{\alpha \in S}$ で表す。更に、その部分集合 S_a および S_A に関する状態確率ベクトルを $\mathbf{p}'_a(t) = [p'_\alpha(t)]_{\alpha \in S_a}$ 、 $\mathbf{p}'_A(t) = [p'_\alpha(t)]_{\alpha \in S_A}$ で表す。

また、同値類 S_a が時刻 t_a で集約された場合、集約された状態を代表状態と呼び a で表し、代表状態の状態確率 $\hat{p}_a(t)$ を、同値類 S_a に含まれる状態確率の和とする。また、代表状態 a から同値類 S_b への推移確率行列を $\mathbf{Q}'_{ab}(t_a) = [q'_{a\beta}(t_a)]_{\beta \in S_b}$ で、代表状態 c から代表状態 a への推移確率を $q'_{ca}(t_c)$ で表す。

次に同値類 S_a の集約手順を示す。

[同値類 S_a の集約手順]

(1) 代表状態 a の状態確率 $\hat{p}_a(t)$ を次式より求める。

$$\hat{p}_a(t) = \sum_{\alpha \in S_a} p'_\alpha(t) \quad (17)$$

(2) 代表状態 a から同値類 S_b への推移確率行列 $\mathbf{Q}'_{ab}(t_a)$ を次式で求める。

$$\mathbf{Q}'_{ab}(t_a) = \frac{\mathbf{p}'_a(t_a)}{\hat{p}_a(t_a)} \mathbf{Q}_{ab} \quad (b = a - 1, a, A) \quad (18)$$

(3) 既に集約されている代表状態 c から代表状態 a への推移確率を次式で求める。

$$q'_{ca}(t_c) = \sum_{\alpha \in S_a} q'_{c,\alpha}(t_c) \quad (c = a, a + 1) \quad (19)$$

続いて、状態確率の和が小さくなった同値類を集約しながら計算を行う方法を示す。

[計算時間の削減を考慮した計算方法]

- (1) $t \leftarrow 0$, $m' \leftarrow N$ とする。
- (2) 状態確率の和がしきい値 ε よりも大きい同

値類の中で添え字が最大であるものを求めこの添え字の値を m とする。但しすべての同値類について、状態確率の総和が ε よりも小さい場合には m を -1 にする。

(3) $m \neq m'$ の場合、 S_a ($a = m + 1, \dots, m'$) について状態集約を行う。

(4) Δt 時間後の吸収状態確率 $\mathbf{p}_A(t + \Delta t)$ を次式により求める。

$$\begin{aligned} \mathbf{p}'_A(t + \Delta t) &= \mathbf{p}'_A(t) + \sum_{b=0}^m \mathbf{p}'_b(t) \mathbf{Q}_{bA} \\ &+ \sum_{b=m+1}^N \hat{p}_b(t) \mathbf{Q}'_{bA}(t_b) \end{aligned} \quad (20)$$

(5) Δt 時間後の過渡状態確率 $\mathbf{p}_T(t + \Delta t)$ を、集約されていない同値類と集約された同値類に分けて求める。

(a) 集約されていない同値類 S_0 から S_m までの状態確率

$$\begin{aligned} \mathbf{p}'_a(t + \Delta t) &= \begin{cases} \mathbf{p}'_a(t) \mathbf{Q}_{aa} + \mathbf{p}'_{a+1}(t) \mathbf{Q}_{a+1,a} \\ \quad (a = 0, \dots, m - 1) \\ \mathbf{p}'_a(t) \mathbf{Q}_{aa} + \hat{p}_{a+1}(t) \mathbf{Q}'_{a+1,a}(t_{a+1}) \\ \quad (a = m) \end{cases} \end{aligned} \quad (21)$$

(b) 集約された同値類の状態確率

$$\begin{aligned} \hat{p}_a(t + \Delta t) &= \begin{cases} \hat{p}_a(t) q'_{aa}(t_a) + \hat{p}_{a+1}(t) q'_{a+1,a}(t_{a+1}) \\ \quad (a = m + 1, \dots, N - 1) \\ \hat{p}_a(t) q'_{aa}(t_a) \\ \quad (a = N) \end{cases} \end{aligned} \quad (22)$$

(6) $t \leftarrow t + \Delta t$, $m' \leftarrow m$

(7) $1 - \sum_{\alpha \in S_A} p_\alpha(t) \geq \delta$ ならステップ (2) に戻り、そうでないなら終了する。

4.2 誤差の上限

前節で示した手順に基づいて得られる極限状態確率の誤差について検討を行う。吸収状態確率の極限値を $\mathbf{p}_A = [p_\alpha]_{\alpha \in S_A}$ で表し次式で定義する。

$$\mathbf{p}_A \equiv \lim_{t \rightarrow \infty} \mathbf{p}_A(t) \quad (23)$$

状態集約を行いながら計算した状態確率の極限値を $\mathbf{p}'_A = [p'_\alpha]_{\alpha \in S_A}$ で表す。

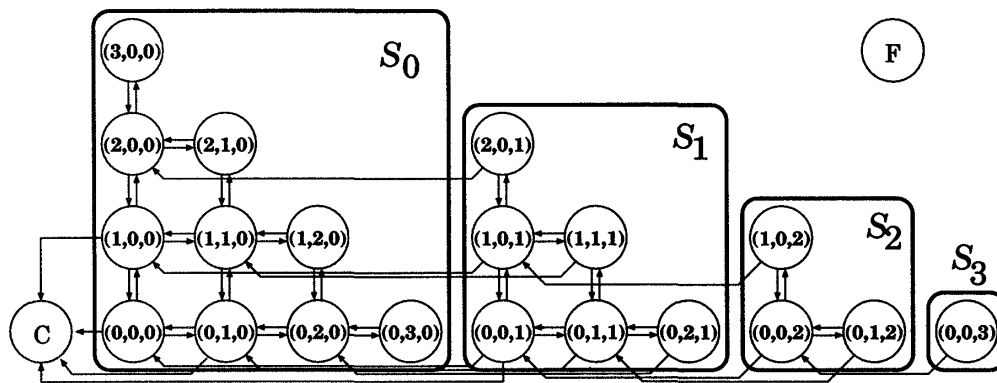


図1 タグ付きタスクモデル (クラス3)
Fig.1 Tagged-task-model (Class 3).

S_a を集約することにより, $\sum_{\alpha \in S_a} p_{\alpha}(t_a)$ だけの確率が集約しない場合と異なる状態へ推移する可能性があり, これが誤差となる. このような誤差の原因となり得る同値類集約時の状態確率の和 p_{err} は, 初期状態における m の値を m_0 で表すと, 次式で与えられる.

$$p_{err} = \sum_{a=m_0+1}^N \sum_{\alpha \in S_a} p_{\alpha}(0) + \varepsilon(m_0 + 1) \quad (24)$$

ここで, 第1項目は初期状態において $\sum_{\alpha \in S_a} p_{\alpha}(0) \leq \varepsilon$ であり集約される同値類の状態確率の和, 第2項目は計算途中で集約される同値類の数としきい値 ε の積である. p_{err} で与えられる確率がすべて集約しない場合と異なる吸収状態に推移した場合に, p'_A の誤差は最大となり, 次式で与えられる.

$$\max_{\alpha \in S_A} \{|p'_{\alpha} - p_{\alpha}|\} \leq p_{err} \quad (25)$$

5. 解析例

5.1 状態の分割

$M = 3, s = 2, L = 4$ であるシステムについて, タスククラスが3であるタスクの処理の過程を解析する例を示す.

連続時間型マルコフ連鎖でモデル化した状態遷移図とその状態集合を到達可能性に基づいて同値類に分割した結果を図1に示す. なお, ここでは状態C以外のすべての状態から生じ得る, タグ付きタスクが捨てられたことを表す状態Fへの推移は省略してある. ここで直接到達可能な同値類の関係は,

$$S_3 \rightarrow S_2, S_2 \rightarrow S_1, S_1 \rightarrow S_0 \quad (26)$$

のみであることから, 行列 Q_{TT} は次のようになる.

$$Q_{TT} = \begin{pmatrix} Q_{00} & 0 & 0 & 0 \\ Q_{10} & Q_{11} & 0 & 0 \\ 0 & Q_{21} & Q_{22} & 0 \\ 0 & 0 & Q_{32} & Q_{33} \end{pmatrix} \quad (27)$$

更に同値類 S_0 を 3.2 に示す方法で分割すると, 図2のような部分集合に分割される. $S_{(0,0)}, S_{(1,0)}, S_{(2,0)}, S_{(3,0)}$ 間で1回の推移で到達可能な部分集合は,

$$S_{(0,0)} \leftrightarrow S_{(1,0)}, S_{(1,0)} \leftrightarrow S_{(2,0)}, S_{(2,0)} \leftrightarrow S_{(3,0)} \quad (28)$$

のみであることから, Q_{00} は次のような帯行列で表される.

$$Q_{00} = \begin{pmatrix} Q_{(0,0)(0,0)} & Q_{(0,0)(1,0)} & 0 & 0 \\ Q_{(1,0)(0,0)} & Q_{(1,0)(1,0)} & Q_{(1,0)(2,0)} & 0 \\ 0 & Q_{(2,0)(1,0)} & Q_{(2,0)(2,0)} & Q_{(2,0)(3,0)} \\ 0 & 0 & Q_{(3,0)(2,0)} & Q_{(3,0)(3,0)} \end{pmatrix} \quad (29)$$

5.2 状態確率と誤差

表1に示すパラメータを用いて, クラス5のタスクの解析を行った結果を示す. ここで ρ はシステムの負荷を表し, 次式で与えられる.

$$\rho = \frac{1}{s} \sum_{j=1}^M \frac{\lambda_j}{\mu_j} \quad (30)$$

また, $b_j (= \nu_j / \mu_j)$ は各クラスのタスクの制限時間の

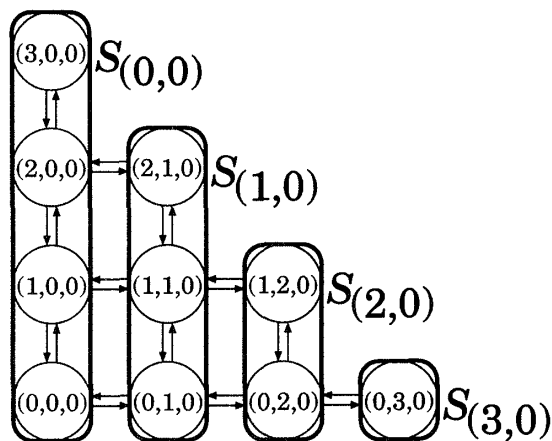


図2 同値類 S_0 の再分割
Fig.2 Dividing coset of S_0 .

表1 システムのパラメータ
Table 1 System parameters.

L	the max. number of tasks	9
s	Number of processors	3
M	Number of task-class	5
ρ	load factor	0.1, 0.2, ..., 0.9
b_j	the normalized mean of deadline	0.01
ε	threshold	0.01, 0.0001

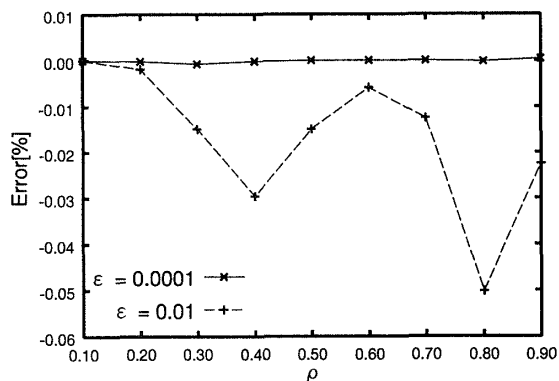


図3 p_C の相対誤差
Fig.3 Relative error of p_C .

厳しさを表す値であり、ここではすべてのクラスについて一定の値 0.01 を用いている。更に、ここではクラス j の処理時間、発生間隔、制限時間に関するパラメータは次式より求めた値を用いている。

$$\mu_j = 0.1^j \quad (31)$$

$$\lambda_j = s\mu_j\rho/M \quad (32)$$

$$\nu_j = b_j\mu_j \quad (33)$$

極限状態確率の相対誤差を図3と図4に示す。 $\varepsilon = 0.0001$ と小さい場合には、0.05% 以内の精度が得られているが、 $\varepsilon = 0.01$ と大きくした場合には最

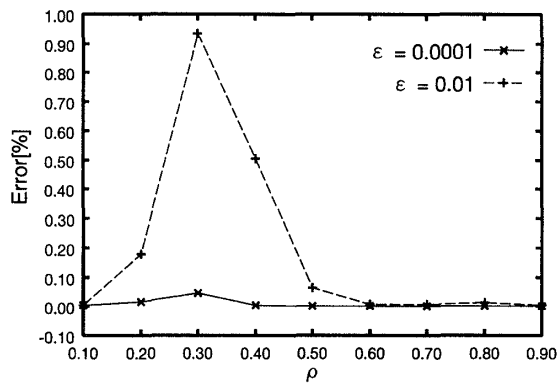


図4 p_F の相対誤差
Fig.4 Relative error of p_F .

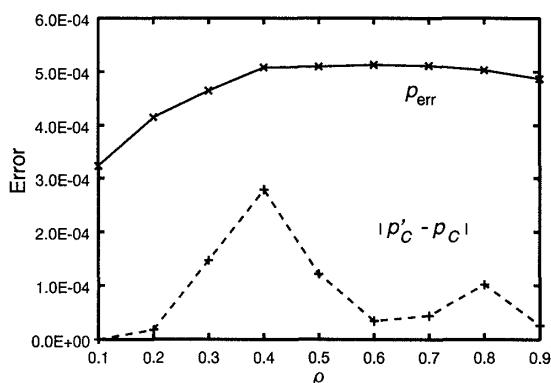


図5 p_C の絶対誤差
Fig.5 Absolute error of p_C .

大で 1.0% となっている。

また、 $\varepsilon = 0.01$ の場合の極限状態確率の絶対誤差と、誤差の上限值 (25) の結果を図5に示す。数値計算結果の誤差 $|p'_C - p_C|$ は p_{err} 以下となっており、式 (25) により、誤差の上限が与えられることがわかる。

また、 $\rho = 0.5$ である場合の計算に要した時間を表2に示す。集約なしはすべての状態を用いて収束するまで状態確率を計算した場合の値である。また、平均状態数は計算過程において各 Δt 刻みの計算を1ステップとしたとき、各ステップでの状態数の和をステップ数で割った値を表す。

計算に要する時間は、極限状態に達するまですべての状態確率を計算する場合に比べて、 $\varepsilon = 0.01$ として集約を行う場合で 25% と大幅に削減されている。

更に、同値類を再分割した場合の部分集合の状態数の最大値に関する結果を表3に示す。再分割を2段階繰り返すことにより部分集合のサイズは 1/10 以下となっており、少ない記憶容量で計算可能であることがわかる。

表2 計算時間 ($\rho = 0.5$)
Table 2 Computation time ($\rho = 0.5$).

ϵ	avg. num. of states	comput. time
without aggregation	1289	99:51.8
0.01	184.16	26:35.2
0.0001	390.33	45:03.1

表3 状態数
Table 3 Number of states.

num. of partitioning	max. subset	num. of states
0	S_0	495
1	$S_{(0,0)}$	165
2	$S_{(0,0,0)}$	45

6. むすび

本論文では、タグ付きタスクモデルの性質に基づいて効率的に状態確率を求める方法を示した。はじめに、状態集合を到達可能性に基づく同値類に分割した後、更に同値類内を分割することによって必要な記憶容量を減らす方法を示した。続いて、タグ付きタスクモデルが少数の吸収状態と多数の過渡状態からなっており、過渡状態の状態確率が時間の経過と共に単調に減少するという性質に基づいて、状態確率の小さな同値類に含まれる状態を集約することによって計算時間を削減する方法を示した。最後にこれらの方法を用いて計算を行った結果を示し、このような考慮をしない場合に比べて本方法を用いた場合には、短時間にかつ十分な精度の解析を行えることを示した。

今後この計算方法を用いて待ち行列ネットワークの解析等を進めていく予定である。

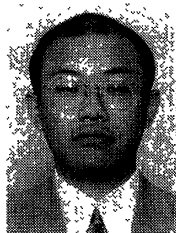
文 献

- [1] B. Berthomieu and M. Diaz, "Modeling and verification of time dependent systems using time Petri nets," IEEE Trans. on Software Engineering, vol.17, no.3, pp.259-273, March 1991.
- [2] K.S. Trivedi, J.K. Muppala, and S.P. Woolet, "Realtime-systems performance in the presence of failures," IEEE Computer, vol.24, no.5, pp.37-47, May 1991.
- [3] B. Melamed and M. Yadin, "Randomization procedures in the computation of cumulative-time distributions over discrete-state Markov processes," Operation Research, vol.32, no.4, pp.926-944, July-Aug. 1984.
- [4] W. Zhao and J.A. Stankovic, "Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems," Proc. of The IEEE 10th Real-Time Systems Symp., pp.156-165, 1989.
- [5] 今井博英, 角山正博, 石井郁夫, 牧野秀夫, 内藤祥雄, "リスポンシブシステムにおけるプロセッサ故障時の解析方法について," 信学論 (D-I), vol.J78-D-I, no.8, pp.716-723,

Aug. 1995.

- [6] 今井博英, 角山正博, 石井郁夫, 牧野秀夫, "タグ付きタスクモデルのためのマルコフ連鎖の一解法," 信学技報, FTS96-52, Dec. 1996
- [7] P.J. Schweitzer, "AGGREGATION METHOD FOR LARGE MARKOV CHAIN," Mathematical Computer Performance and Reliability, pp.275-286, 1984.
- [8] P. Semal, "Refinable bounds for large Markov chains," IEEE Trans. on Computers, vol.44, no.10, pp.1216-1222, Oct. 1995.
- [9] K.S. Trivedi and A. Bobbio, "An aggregation technique for the transient analysis of stiff Markov chain," IEEE Trans. on Computers, vol.C-35, no.9, pp.803-814, Sept. 1986.
- [10] 宮沢政清, "確率と確率過程," 近代科学社, 1993.
- [11] 田口玄一, "確率・統計," 日本規格協会, 1992.

(平成9年2月10日受付, 4月24日再受付)



今井 博英 (学生会員)

平5新潟大・工・情報卒。平7同大学院修士課程了, 現在同大学院自然科学研究科博士後期課程在学中。リアルタイムシステム, フォールトトレラントシステムの研究に従事。



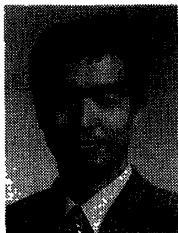
角山 正博 (正員)

昭44新潟大・工・電子卒。同年東京芝浦電気(株)入社。昭46横河ヒューレットパッカード(株)入社。主に測定器の研究開発に従事。長岡技術科学大学教務職員, 助手, 長岡工業高等専門学校教授を経て, 現在新潟工科大学教授。論理回路の故障診断, 高信頼性並列処理システム, リアルタイムシステム等に興味をもつ。平2工博。情報処理学会, IEEE等各会員。



石井 郁夫 (正員)

昭38新潟大・工・電気卒。同大・電子助手, 助教授, 同大・情報教授を経て, 現在同大学院・自然科学研究科教授。この間, 画像・図形処理等の研究に従事。現在人工現実感, ヒューマンインタフェース, 画像・図形処理装置, リアルタイムシステムなどの研究を行っている。工博(東工大)。情報処理学会, 日本ME学会各会員。



牧野 秀夫 (正員)

昭51新潟大・工・電子卒。昭53同大大学院修士課程了。同年情報勤務。平2助教授、平7教授。現在に至る。平1より1年間カナダ・トロント大にて生体信号処理の研究に従事。最近は、埋め込み型医用機器、福祉機器、画像処理、リアルタイムシステムなどの研究を行っている。工博、IEEE、日本ME学会各会員。