

AISにおける送信予約システムのPRISMを用いた解析と検証

豊島崇士 高橋和子

船舶自動識別システム (AIS : Automatic Identification System) は船舶の安全な航海を目的とした通信システムだが、他の船舶の情報が正しく送受信できない場合が存在する。本研究では AIS の一部である送信予約システムに対し、確率付きモデル検査器 PRISM を用いて正しく送受信できない場合の条件とその確率を解析し、その結果を元に検証を行い確率的に安全であることを示す。

1 はじめに

2000 年、国際海事機関 (IMO : International Maritime Organization) の海上安全委員会は「海上における人命の安全のための国際条約」(SOLAS 条約 : Safety Of Life At Sea Convention) を改訂し、その結果、平成 14 年 (2002 年)7 月から平成 20 年 (2008 年)7 月までに、データ通信機能を具備した AIS の旅客船舶や外航の貨物船舶、漁船舶などへの順次搭載が義務化された。AIS とは船舶に搭載することで識別符号、船舶名、位置、針路、船舶の速度といったその船舶の航海に関する情報を自動的かつ周期的に他の船舶や陸上の施設へ送信、他の船舶からの情報の受信をする装置のことである [1][2]。

AIS は安全面などで非常に役に立つシステムであるが、船舶への搭載の義務化も最近のことであり、まだ新しいシステムである。そのため、AIS 内部のシステムが不具合なく動作するかどうかという検証は行

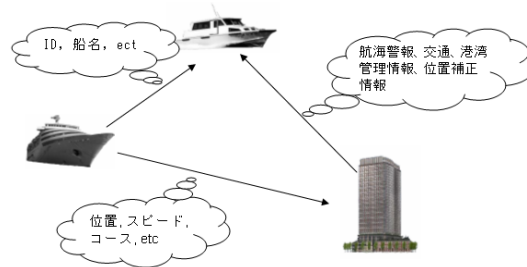


図 1 AIS の働き

われていない。

AIS の通信方式では同時に 2 隻以上の船舶が情報を送信しようとする時、その情報が他の船舶に届かない状況になる。そこで、AIS では情報を送信すると同時に、次に送信する時間の予約を行う。予約を行うことで、同時に多数の船舶が情報の送信を行うことがなくなる。

しかし、その予約システムに対して検証が行われた例がなく、多数の船舶が同じ時間に対して送信の予約してしまった結果、同時に情報の送信を行う場合がないとは言いきれない。そこで、本論文ではこの送信予約システムに着目し、数理的技法の 1 つであるモデル検査技術を使ってシステムに不具合がないことを示すことを目的とする。

我々はモデル検査器 SMV [6] を用いて「予約が重なることなく取れるか」についての検証を行った。その結果、実際の AIS の予約の取り方では 2 隻以上の予約が重なる場合が必ず存在することがわかった [15]。実際の AIS では複雑なプロトコルによって安全を保

Takashi Toyoshima, Kazuko Takahashi, 関西学院大学, Kwansai Gakuin University

証するよりも、確率的に安全であれば容認されている。なぜなら予約できる時間は船舶の数に対して非常に多く、2隻以上の船舶が同じ時間に対して予約を行うことは確率的にまずないためである。さらに、予約の重なりが起こったとしても、同じ時間に対して予約し続けるわけではない。しばらく時間が経過すると別の時間に予約を行い、その時に予約の重なりが解消されるため、1度くらいの予約の重なりは無視できるとされていることも理由として挙げられる。SMV で出力される反例を見ても、予約スロットの重なる場合は非常に稀であることもわかったが、どの程度の確率で安全なのか明確にはわからなかった。

そこで、本論文では「予約が重なることなく取れるか」が確率的に安全であるという仕様を対象とし、確率付きモデル検査器 PRISM [7][8] を用いて確率計算と解析および検証を行う。

実際に AIS が行っている送信予約システムは複雑すぎてそのままモデル化すると実行時に状態爆発が起こる。そのため本論文では、以下の3点を満足するように簡単化したモデルをいくつか作成し、それぞれについて安全性を調べ、検証を行った。

- 他の船舶に予約されている時間は予約しないように避ける
- 基本的に予約は周期的に行う
- 確率性を加味する

また、それらの予約の決め方と比較するためのモデルとしてランダムに予約を選択する方法を取るモデルも作成した。

作成したモデルを PRISM で動かし、仕様を満たす確率の算出および検証を行ったところ、結果として実際の AIS での予約の取り方は予約時間が密集していない場合は予約が重なる確率が低く安全であるとわかった。また、密集している場合は確率が決して低くはないが、少し時間が経つとほぼ必ず予約が重ならず送信が正しく行われることがわかった。

本論文の構成は以下の通りである。2節で AIS の送信予約システムを説明する。3節では時相論理やモデル検査ツールについてなどモデル検査に関する事柄を述べる。4節では AIS の送信予約システムのモデル化を行い、5節で作成したモデルを PRISM で動

かし、解析と検証を行う。第6章ではまとめと今後の課題について述べる。

2 システムの説明

モデル検証の対象とする AIS で用いられている送信予約システムについて説明を行う。

2.1 AIS における送信予約

AIS は他の船舶や陸上の施設と情報を交換し合う通信機能を持つが、AIS の送受信機はいわばトランシーバーのようなもので、自船舶が情報の送信を行っている時に他の船舶が送信した情報が受信できない状態になってしまう。つまり、同時に2隻以上の船舶が送信を行ってしまうと、それらの船舶は互いの情報を得ることができなくなるといった、正しく情報が受信されているか確かでない場合が考えられる。AIS ではそういった状況を改善するために、自己管理型時分割多元接続 (SOTDMA : Self Organized Time Division Multiple Access) 方式という通信方式がとられている。この方式ではまず1分を1フレームと定義し、そのフレームを2250のタイムスロットに分割する。1スロットの長さは26.7msとなり、このスロットにあたる時間に情報を送信するのだが、その送信を行うのと同時に次にどのスロットで情報送信を行うかもその情報の中に含ませ、次に送信するスロットの予約を行う。

このフレームには2種類あり、1つは既に全ての船舶が予約済みで予約スロットの決定や情報の送信する時間を確認するための現在進行中のフレームで、もう1つは次の情報の送信を行うための予約を行うフレームである。各船舶はこれら2つのフレームをそれぞれ持っている。船舶 S は現在のフレームを確認し、予約していたスロット時間になると次の予約スロットを決め、自船舶のフレームに予約したことを書き込み、他の船舶に情報を送信する。その情報を受信した他の船舶は船舶 S が予約したスロットの箇所では情報を送信するという自分の船舶がそれぞれのフレームに書き込む。

例えば、船舶 A、船舶 B、船舶 C が通信する場合を想定する。船舶 A は、現在のフレームで予約をし

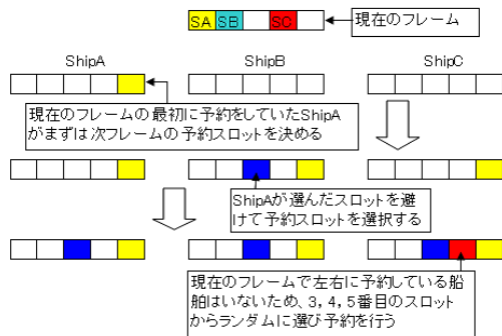


図2 予約スロットの決め方

ていたスロット時間になると現在のフレームで予約をしていたスロット箇所と同じスロットか、そのスロットの左右のスロットの3箇所の中の他の船舶が予約を行っていなかったスロットをランダムに1つ選択し、それを次のフレームでの予約スロットとする。船舶Aは予約スロットが決まれば、自分の次フレームにその情報を書き込み、他の船舶へ送信を行う。船舶B、Cはそれぞれ受信した船舶Aの予約スロット情報を自分の次フレームに書き込んでいく。船舶B、Cも同様に現在のフレームを確認してから3つのスロットのうちで他の船舶が予約を行っていなかったスロットをランダムに1つ選び、それを予約スロットとして自船舶の次フレームに書き込み、他の船舶へ送信といったように予約スロットを決めていく。この結果、1つのフレームが終了すると次のフレームになる。これらを次々に繰り返していき、それぞれの船舶が、スロットが衝突しないように、自船舶の情報とスロットの予約情報を送信しながらお互いに通信する。また、予約スロットを決定する際には、基地局という陸上施設が設定した何秒間隔(何スロット間隔)で情報を送信するという条件がついており、基本的には一定周期で情報を送信するようになっている。

2.2 パースト状態における問題点

この方式では、各船舶が空きスロットを確認してから次フレームに予約を入れるまでにはタイムラグが発生する可能性がある。そのため、複数の船舶が空きスロットと思って同一のスロットに予約を入れてしまう

場合がある。たとえば図3の例を考える。

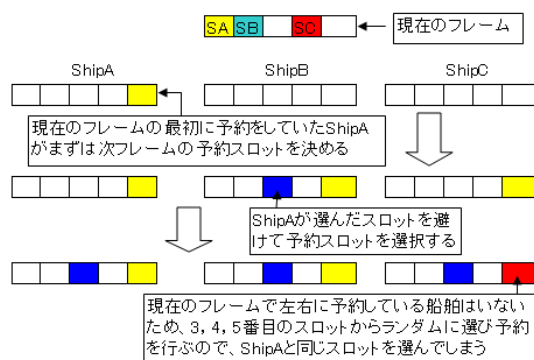


図3 パースト状態に陥る例

船舶A、B、Cが通信する場合を想定する。船舶Aが予約を行い、その後船舶Bが船舶Aが現在のフレームで選択していたスロットを避けて予約スロットを選ぶ。船舶Bは自分のフレームに選んだスロットを書き込み、他の船舶へ送信を行う。続けて船舶Cが予約スロットを決めるわけだが、船舶Aと同じ予約スロットを選ぶ可能性がある。その結果、各船舶のフレーム状況は図3のようになり、船舶A、Cは同じスロット時間に情報の送信を行ってしまうことになる。

送信が重なった場合、送信を行った船舶同士は相手の情報を受信することができないが、そのときに発生する問題は情報が受信できないという問題だけではなく他にもある。船舶の情報の送信が重なってしまった場合、受信ができていないのはそれらの船舶だけではなく、周りにいる受信状態であった船舶もまた送信が重なってしまった船舶の情報を得ることができないパースト状態に陥る。

この状態になった場合に、周辺で受信をしようとしていた船舶にはその時間になにかしら情報の送信が行われ、重なってしまったことはわかるのだが、内容が分からないためどの船舶がパーストしたかなどは一切分からない。さらに、送信が重なってしまった当事者の方は重なっているか重なっていないかは関係なく送信を行うので、パーストしていてもそれがわかっていない。つまり、パースト状態になると予約スロットを修正することができないことになる。

情報送受信が行われずにそのままではとても危険な状態であるため、バースト状態は回避しなければならない。

3 モデル検査

ここではモデル検査および確率付きモデル検査器 PRISM について述べる。

3.1 モデル検査とは

一般に、システムの不具合があるかないかを調べる際に、テストデータを用いたシミュレーション検査が行われるが、その方法では検証するのに時間的コスト、人的コストの問題がある。テストデータを使わず、あらゆる状態や入出力に対して自動的に検査する手段として注目されているのがモデル検査である。モデル検査ではシステムの有限状態モデルとそのシステムが満たすべき仕様として記述された論理式を与えられた時、検証過程ではシステムがその仕様を満たしているかどうかを網羅的に検査する。仕様は論理を使って書かれることが多い[3]。

一般的に知られているモデル検査器としては SMV や SPIN [10], UPPAAL [11] 等が挙げられるが、本研究では確率を含んだシステムの動作をモデル化、検証を行うために確率付きモデル検査器を用いることにする。確率付きモデル検査器には、APMC [12], ETMCC [13], YMER [14] などがあるが、本研究では GUI の使いやすさ、ユーザの多さの点から PRISM [7][8] を用いることにする。

3.2 PRISM とは

PRISM はバーミンガム大学と現オックスフォード大学が開発した確率付きモデル検査器である。PRISM を用いることで確率を含んだ動作を形式的にモデル化し、その動作の解析と検証を行うことが可能である。PRISM では 3 タイプの確率モデルに対応しており、それぞれ

- 離散時間マルコフ連鎖: discrete-time Markov chains(DTMC)
- 連続時間マルコフ連鎖: continuous-time Markov chains(CTMC)

- マルコフ決定過程: Markov decision processes(MDP) となっている。

「将来における事象の起こる確率は、現在の状態だけに依存し、過去の経歴にはよらない」という性質をマルコフ性といい、確率的に変動する過程がマルコフ性を満たすとき、マルコフ連鎖と呼ぶ。その中で、時間が離散的なものを DTMC、時間が連続的なものを CTMC、状態が確率的にマルコフ連鎖に従って推移する多段階決定過程を MDP という。本研究では DTMC を使用する。

また、PRISM で検証に用いられる時相論理は PCTL [4] と CSL [5], LTL の 3 つがあり本研究では仕様を PCTL で記述する。

3.3 PRISM における PCTL

PCTL(Probabilistic real time Computation Tree Logic) は CTL(Computation Tree Logic)[3] を拡張し、確率的な検証を行えるようにした時相論理の一つである。

PCTL の構文規則は以下ようになる。

$$\phi ::= \alpha \mid !\phi \mid \phi_1 \& \phi_2 \mid \phi_1 \text{ '}' \phi_2 \\ \mid \phi_1 \quad \phi_2 \mid P_{\beta p} [\psi]$$

$$\psi ::= X\phi \mid \phi_1 U^{\leq \gamma} \phi_2 \mid G^{\leq \gamma} \phi \mid F^{\leq \gamma} \phi$$

ϕ は state formula, ψ は path formula を表している。 α は原子命題, p は確率を表しており範囲は $p \in [0,1]$, β は関係演算子を表し $\beta \in \{\leq, <, >, \geq\}$ である。

X, U, G, F は時相演算子と呼ばれるもので、CTL と同様単一パスにおける検査を記述するのに使用する。

また、PCTL では有限回遷移した場合と、無限回遷移した場合いずれにおける確率も求めることができる。PRISM では状態から状態への遷移を 1 ステップとしており、PCTL 式において X, F, G, U のような時相演算子を使用する際には、 γ に有限回のステップを表す特定の自然数を入れる。また、それぞれの演算子の後の $\leq \gamma$ を取りはずすと無限回遷移する場合が表現できる。

ψ に含まれる時相演算子を用いた論理式を path property といい、それぞれの意味は以下の通りで

ある .

- $X\phi$: "Next" path properties
ある状態の次の状態が ϕ を満たすことを表す . X ではステップ数が 1 と決まっているので γ を指定する必要はない .
- $\phi_1 U^{\leq \gamma} \phi_2$: "Until" path properties
ある状態から γ ステップ以内の状態において ϕ_2 が満たされていると , それ以前の状態で ϕ_2 が満たされていることを表す .
- $G^{\leq \gamma} \phi$: "Globally" path properties
ある状態から γ ステップまでの状態全てで ϕ が満たされていることを表す .
- $F^{\leq \gamma} \phi$: "Eventually" path properties
ある状態から γ ステップまでのどれか状態 ϕ が満たされることを表す .

PRISM ではある式の起こる確率を求めたり , 確率付きで書かれた仕様を検証することができる . $P_{\beta p}[\psi]$ は ψ が成り立つ確率が βp であることを表している . 例えば , 「ある状態 s から 10 ステップ経過するまで ϕ が成り立ち続ける確率は 25 % 以下である」という仕様は $P_{\leq 0.25} [G^{\leq 10} \phi]$ となる . ここである状態 s とは , 特に指定がない限り PRISM では全ての状態とみなされる . つまり , 検証式の意味は 「全ての状態において 10 ステップ経過するまで ϕ が成り立ち続ける確率は 25 % 以下である」となる .

全ての状態からではなく初期状態から検証を開始したい場合には "init" => $P_{\leq 0.25}$ と記述し , ϕ が成り立つ状態から検証を始めたい場合には $\phi => P_{\leq 0.25}$ と記述する必要がある .

また , PRISM では検証だけではなく仕様を満たす確率も求めることが可能である . $P=?[\psi]$ と記述すれば , ψ が成り立つ確率がいくらかを計算することができる . 例えば 「ある状態 s から 10 ステップ経過するまでに ϕ が成立する確率はいくらか」という記述は $P=? [F^{\leq 10} \phi]$ となる . ここでいうある状態 s も上記の検証の時と同様に , 特に指定しなければ全ての状態が対象となる . そのため , 確率の計算は全ての状態から行われ , 全ての状態に対して 「10 ステップ経過するまでに ϕ が成立する確率」が計算される . 特定の初期状態における確率を求めたい場合には $P=?$

$[F^{\leq 10} \phi \{ "init" \}]$ と記述し , ϕ' が成り立つ状態における確率を求めたい場合には $P=? [F^{\leq 10} \phi \{ \phi' \}]$ と記述する必要がある .

3.4 PRISM を用いた確率計算と検証の注意点

PRISM における計算は慣れないものには直観的に分かりづらい面がある . 図 4 のように遷移に確率を付与された状態遷移図に対し PRISM を用いた確率計算と検証を考える .

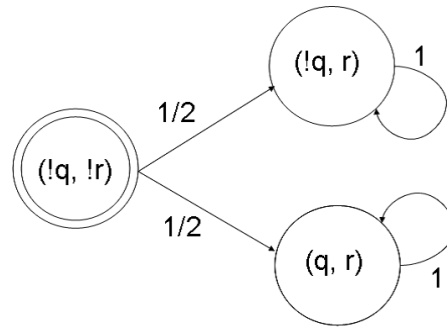


図 4 PRISM の解析及び検証例

ex1 $P_{\geq 0.5} [G^{\leq 2} q]$

ex2 $P_{\leq 0.5} [G^{\leq 2} q]$

それぞれの式は $q=TRUE$ となる状態が 2 連続で満たされる確率が ex1 の式では 0.5 以上 , ex2 の式では 0.5 以下であるかことを示す . この 2 つの式を PRISM で検証したところ , どちらの結果も FALSE となり仕様を満たす確率は 0.5 以上でも 0.5 以下でもないことになる . この検証結果は直観に反するが決して間違いではない . PRISM では全ての状態が ex1 , ex2 の式を満たすか調べているため , この状態遷移図だと (!q, r) を初期状態とすると $[G^{\leq 2} q]$ となる確率は $P=0$ となり , (q,r) からだと $P=1$ となる . いずれの式も 「全ての状態において」という条件を満たしていないため , 結果は ex1 , ex2 ともに FALSE となる .

検証ではなく確率を求める例として以下の式を考える .

ex3 $P=? [G^{\leq 2} q]$

この式の意味は「2ステップ連続で q が満たされる確率はいくらか」である。結果としては $P=0.5$ という確率が出力される。PRISM において確率を求める場合には全ての状態における確率を独立に算出しており、結果に出力される確率は1つだけ代表をランダムに選んだものである。0.5 という結果は初期状態を $(!q,!r)$ とした場合の結果であり、全ての状態の確率計算結果は log ファイルに格納されている。

また、PRISM における PCTL では CTL のように時相演算子 X, G, F, U を組み合わせて検証式を記述することも可能である。例えば

$$\text{ex4 } P_{\geq 0.5} [G^{\leq 1} (q \ \& \ P_{\geq 0.5} [X \ q])]$$

のような式が記述できる。この式の意味としては「初期値から2連続で $q=TRUE$ を満たし、さらに $q=TRUE$ を満たす状態の次の状態でも $q=TRUE$ を満たす確率は0.5以上である」となるように見える。

しかし、実際は少し違っており「初期値から2連続で $q=TRUE$ が満たされ、かつその次の状態が0.5以上の確率で $q=TRUE$ が満たされている確率が0.5以上である」となる。

4 送信予約システムのモデル化

本研究では予約スロットの決め方に重点を置き、実際の AIS での予約スロットの決め方に出来るだけ即した方法を用いて「選定モデル」を4つ作成した。以下のリストにある prob1 が本来の AIS に1番近い予約を行うモデルであり、それ以外の3つは予約スロットの決め方の条件を少し変化させたモデルである。

prob1 隣1つを調べ、空いている方を高確率で選ぶ。

det1 隣1つを調べ、空いている方を必ず選ぶ。

prob2 隣2つを調べ、空いている方の隣を高確率で選ぶ。

det2 隣2つを調べ、空いている方の隣を必ず選ぶ。

それぞれのモデルに対し、詳しい説明を行う。

4.1 選定モデル

実際の AIS では基本的に現在のフレームで選択していたスロットと同じスロットを次のフレームでも選

択をするが、数回に1回、左右のスロットのどちらかを選択するという方法を取っている。また、左右のスロットを選択する際には現在のフレームの左右のスロットを確認し、他の船舶が予約をしているスロットは選択しないようにしている。

そこで、選定モデルでは実際の AIS の予約スロットの選択方法に近づけるために、次のように予約スロットを選択するようにした。

まず船舶は現在のフレームで予約していたスロット箇所の左右のスロットで既に予約されている数を左右それぞれ調べ、少ないほうを80%の確率で選択、もう片方の隣と現在のフレームで予約しているスロットを10%の確率で選択するとした。また、数が同じ場合には3つのスロットを完全にランダムに選択することにした。確率で予約スロット選択を行うことで、実際の AIS で行われている「数回に1回スロットを選択する」という動作を表現している。

また、このモデルに対して細かく条件を分けて調べることにする。具体的には、

- スロット選択時の確率を変更

上記の方法では数の少ない方のスロットを80%の確率で選択を行っていたが、空いている方のスロットを100%選択する場合についても調べる。これを調べることで、必ず選ぶ場合と多少確率要素を含めた場合との比較を行う。

- 調べるスロット数を変化

確認するスロット数を両隣1つずつと両隣2つずつの2通りに分けて調べる。こうすることで船舶の予約が密集しているフレームのスロット箇所がより早くわかり、予約が密集していない場所を選びやすくなるのではないと思われる。

4.2 random モデル

また、比較対象としてランダムに予約スロットを決めるモデル (random) も作成した。

random モデルでの予約スロットの決め方は現在のフレームで予約したスロット時間になると、現在のフレーム状況は確認せず、1フレームの全スロットの中からランダムに1つ選択し、そのスロットに予約を行うものである。random モデルでは予約スロットを

全くのランダムで決めるため、フレームのロット数が増えれば増えるほど2隻以上の船舶が同じロットに予約をする確率は低くなる。しかし、同じロットを選ぶ確率は0にはならないことが問題である。

5 解析と検証

船舶の数は3隻、1フレームのロット数は10と15とし、ロットには左から1, 2, ... と付番する。作成したモデルをPRISMで動かして、確率解析と検証を行う。本論文で扱うPRISMのバージョンはPRISM3.3beta1である。

複数の船舶が同じロットを選び予約ロットの衝突が起こったことを”bst”と表し、予約ロットの衝突が起こった回数を”bst_sum”と表記する。3隻の船舶の初期予約ロットをあらかじめ決めて初期状態とし”init”と表す。”init”を変化させ、確率がどう変わるかを調べる。

5.1 確率の計算と解析

予約ロットの重なる確率に関する仕様を3つ用意し、それぞれPRISMを用いて確率の計算を行う。それぞれの解析結果を以下に記す。

5.1.1 確率の計算と解析その1

予約ロットが重なることはできるだけ避けたほうがよい。そこでまず、「20フレーム経過するまでに予約ロットの衝突が起こる回数が5回以上である」の確率を求める。PCTL式で表すと次のようになる。

Spec1.1 $P=? [true \ U^{\leq 20frame} bst_sum \geq 5 \ \{ "init" \}]$

予約ロットの数が10の場合の結果は図5のようになった。図5の縦軸は確率を表し、横軸は作成したモデルが並んでいる。縦棒はそれぞれ初期予約ロットを変化させた結果であり、3隻の船舶の初期予約ロットの並びを(S1, S2, S3)とすると、左から(1, 2, 3), (1, 3, 5), (1, 3, 7), (1, 4, 7)と並んでいる。

randomモデルでは初期予約ロットに関係なく、0.7とかなり高い確率になったが、選定モデルでは全般的にかなり低めの確率を得ることができた。初期予約ロットを互いに離せば離すほど確率が下がること

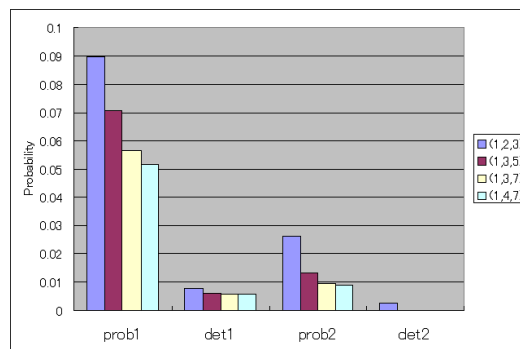


図5 確率計算結果1

から、予約ロットが密集していると予約の重なりが起りやすくなることを示している。また、選定モデルの中でも確率的に予約ロットを決めるより決定的に決めたほうがよいことと、調べる両隣のロット数は多いほうがよいことがわかった。しかし、両隣を調べる範囲を広げすぎることが必ずしも確率を下げるとは限らない。

例えば、ある船Aが予約ロットを決める際に、左1つ隣に他の船舶が予約をしていて、右2, 3つ隣に別の船舶が予約しているとする。そこで、隣3つまで調べてしまうと、左隣を選んでしまうことになるが、左隣の方がより近くに予約していて、予約ロットの重なりが起ってしまう可能性が高いはずである。よって、あまり遠くまで調べすぎると効果が薄れてしまう可能性がある。

なお、ロット数が15の場合には、10の場合よりも全般的に低い確率が得られた。

5.1.2 確率の計算と解析その2

情報のやり取りを行う際に、予約ロットの重なりが起り続け連続で船舶の情報を受信できない時間が長くなってしまうと、AISの利点の1つである船舶同士の衝突回避等ができなくなってしまう。そこで、「予約ロットの衝突が起ってから2フレーム連続で予約ロットが衝突し続ける」の確率を求める。PCTL式で表すと次のようになる。

Spec1.2 $P=? [G^{\leq 2frame} bst \ \{ "bst" \}]$

この式では状態”bst”になった時から2フレームの間に予約ロットが重なるかを調べているため、予約

スロットの初期値に依存しない．よってこのモデルに対しては初期予約スロットをランダムに決定する．

予約スロットの数が 10 の場合の結果は図 6 のようになった．図 6 の縦軸は確率で横軸はモデルを表している．

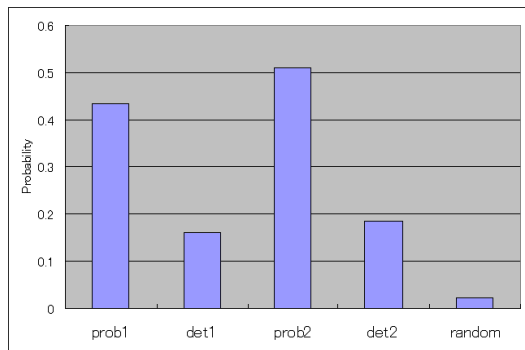


図 6 確率計算結果 2

選定モデルでは、同じスロットに予約した船舶は次にスロットに予約する場合は、隣のスロットのどちらかが他の船舶の予約数が少ないならほぼ同じスロットを選び、また、隣のスロットの船舶の予約数が同じ場合も、同じスロットを選ぶ確率は高めである．そのため、ランダム性があまりないことから連続で予約スロットが重なる確率は高いという結果となった．逆に random モデルは他の船舶の予約状況に関係なく、さらにランダムに取れるスロットの数が多いため、結果として低い確率を得ることになった．

また、スロット数が 15 の場合は、スロット数が 10 の場合と比べてどのモデルも確率はあまり変化がなかった．

したがって、この仕様に関してはランダムな方法が優位であるといえる．

5.1.3 確率の計算と解析その 3

3 つの目の仕様は「10 フレーム経過するまで一度も予約スロットの衝突が起こらない」の確率を求める．PCTL 式で表すと次のようになる．

Spec1.3 $P=? [G^{\leq 10 \text{frame}} ! \text{bst}\{\text{"init"}\}]$

予約スロットの数が 10 の場合の結果は図 7 のようになった．

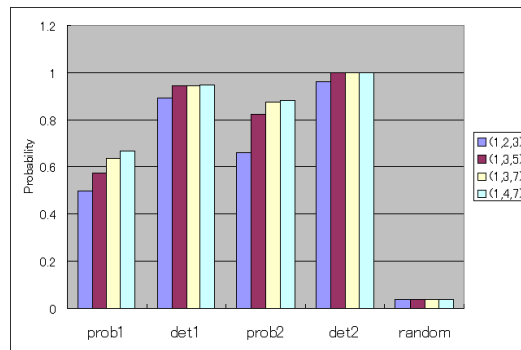


図 7 確率計算結果 3

結果を見ると、random モデルでは確率が 0.1 未満と出力され、10 フレーム中 1 回以上はほぼ確実に予約スロットの重なりが発生していることがわかる．

選定モデルはこれよりもずっと高確率で予約スロットの回避が行われている．調べる隣接スロットの数については、2 つの方が 1 つよりも確率が高いことがわかる．これは広範囲を確認し、予約されているスロットの密度が高い箇所を早期に発見でき、回避が早く行えるためだと思われる．また、スロット数が 15 の場合は、わずかながら全体的に確率の増加が見られ、予約スロットの密度の薄まりが衝突回避に繋がることわかる．

5.2 検証

次に解析によって得られた結果を元に検証を行う．確率の計算、解析である程度予約スロットの重なる確率を求めたが、初期状態を絞った限定的な確率であるため、かならずしも結果が全ての状態に当てはまるとは限らない．検証は全ての状態において仕様を満たすかを調べることができる．検証では船舶の数は 3、スロット数を 10 とした．

検証を行う仕様は 3 つとし、それぞれの仕様、検証結果を以下に記述する．また、以下に記述している仕様の中の $N_i \text{frame} (i=1, 2, 3)$ はフレーム数を表しており、検証時には 3 フレームと 20 フレームの 2 つを使用する． $X_i (i=1, 2, 3)$ は確率を表し、検証段階で任意の値をいれ検証を行う．

5.2.1 検証その 1

最初の検証では「 N_1 フレーム中に 1 回以上は他の船舶と予約が重ならずに、情報の送信が正しく行える確率は X_1 以上である」について検証を行う。PCTL 式で表すと次のようになる。

$$\text{Spec2.1 } P_{\geq X_1} [F^{\leq N_1 \text{frame}} ! \text{"bst"}]$$

確率 X_1 を変動させ検証を行った結果、表 1 のようになった。表の縦にはモデルが並んでおり、横は検証するフレームの長さである。表の値は確率 X_1 の値であり、TRUE の列は仕様が TRUE になる X_1 の最小値、FALSE の列には仕様が FALSE になる X_1 の最大値をそれぞれ表す。

	$(N_1=3)$		$(N_1=20)$
	TRUE	FALSE	TRUE
prob1	0.78	0.79	0.99
det1	0.71	0.72	0.99
prob2	0.74	0.75	0.99
det2	0.71	0.72	0.99
random	0.98	0.97	0.99

表 1 Spec2.1 検証結果

random モデルは 3 回中 1 回以上予約が重ならない確率が 0.98 以上であり、全ての状態で安定して高確率であることがわかった。それに対し、選定モデルでは確率は 0.71 以上となり、少し低めの確率となる状態がそれぞれ存在することがわかる。

選定モデルの中では確率的に選択する方が若干優位である。Spec2.1 では全ての状態を初期値として仕様を満たすか調べているので、予約スロットの重なりがある状態や船舶の予約が密集している場合も初期状態に含んでいる。これらの状態から予約スロットの重なりを回避するためには、random モデルのような選び方をする方がよく、その選び方に近い prob が確率的によりよい結果になったのだと思われる。

さらに、この仕様ではわずかな差ではあるが隣接するスロットは 2 つ調べるより 1 つだけ調べた方がよいこともわかった。

また、20 フレームの場合には、どのモデルにおいても全ての状態において確率は 99% 以上となりほぼ確実に予約スロットの重なりがなく 20 フレーム経過

するまでに一度は送信が成功することがわかる。

5.2.2 検証その 2

Spec1.2 で予約スロットが重なってから連続で重なり続ける確率を求めたが、その結果として出力される確率はいくつかの予約スロットが重なっている状態の内の 1 つから連続で重なる確率を求めたにすぎない。そこで、全ての予約スロットが重なっている状態に対して検査を行うために「予約の重なりが起こってから N_2 フレーム連続で、予約スロットの衝突が起こる確率は X_2 以下である」について検証を行う。PCTL 式で表すと次のようになる。

$$\text{Spec2.2 } \text{"bst"} \Rightarrow P_{\leq X_2} [G^{\leq N_2 \text{frame}} \text{"bst"}]$$

確率 X_2 を変動させ検証を行った結果、表 2 のようになった。

	$(N_2=3)$		$(N_2=20)$
	TRUE	FALSE	TRUE
prob1	~0.25	0.24~	1~0.01
det1	~0.29	0.28~	1~0.01
prob2	~0.29	0.28~	1~0.01
det2	~0.29	0.28~	1~0.01
random	~0.02	0.03~	1~0.01

表 2 Spec2.2 検証結果

フレーム数が 3 の場合には random モデルでは確率が 0.02 以下であるのに対し、選定モデルは 0.29 以下と確率が少し大きいという結果となり、random モデルが優位であることがわかった。また、選定モデルの中ではわずかではあるが、確率的に選択し、確認する隣接スロットの数が 1 つである prob1 が一番よい結果を出した。この仕様に対しては調べる隣接するスロットの数はあまり関係なく、衝突を回避するためによりランダムな選び方をする方がよいといえる。

Spec2.2 ではそもそも仕様自体で予約スロットの重なりがある状態を初期状態としている。そのため、Spec1.2 の時と同様の理由から random モデルおよび、prob1, 2 がよい結果を残すことになった。

また、20 フレームの場合には、確率は 0.01 以下と限りなく低くなっており、予約スロットが重なったとしてもそのうち予約スロットの重なりがなくなり、正しく情報の送信が行えるようになることがわかる。

5.2.3 検証その3

Spec2.1, Spec2.2 ではそれぞれ予約スロットが重なっている状態を含めた状態全てから検査を行っていた。しかし, Spec2.1 から実際には予約スロットの重なっていることはほとんどないはずであるので, 次に検査する仕様は「予約スロットの重なりがない状態から N_3 フレーム中に 1 回以上は他の船舶と予約スロットが重なってしまう確率は X_3 以下である」とした。PCTL 式で表すと次のようになる。

Spec2.3 $! \text{ "bst"} \Rightarrow P_{\leq X_3} [F^{\leq N_3 \text{ frame}} \text{ "bst"}]$

X_3 を変動させ検証を行った結果, 表 3 のようになった。

	$(N_3=3)$		$(N_3=20)$	
	TRUE	FALSE	TRUE	FALSE
prob1	~0.4	0.39~	~0.75	0.74~
det1	~0.21	0.2~	~0.3	0.29~
prob2	~0.21	0.2~	~0.45	0.44~
det2	~0.13	0.12~	~0.12	0.11~
random	~0.63	0.62~	~0.999	0.99~

表 3 Spec2.3 検証結果

予約スロットの衝突している状態を含まない場合は random モデルよりも選定モデルの方がよい結果を出している。また, 選定モデルでも予約スロットの選択に確率要素を含めている prob はあまり選定モデルの中ではあまりよい結果とならず, この仕様に関してはランダムに選択しない方がよいことがわかった。調べる隣接スロットの数は 2 つ調べた方がより確率が低く, この理由としては, 広い範囲を確認することで早期に予約スロットの密集を感知し, 密集している方向を選ばないように予約をしているためだと思われる。

また, 20 フレームの場合には, 決定的に予約スロットを選択する det1, det2 に関しては, 3 フレームの場合とほぼ同じ確率であることから, 初期値からすぐに衝突が起こる状態以外はほぼ 20 フレームが経過しても衝突が起きないことがわかる。しかし, 確率要素を含めた prob1, prob2 および random モデルはフレーム数を増やすと大幅な確率増加が見られた。確率的に予約スロットを決めることは, そのうち予約スロットの重なりが発生することに繋がると思われる。

5.3 全体としての評価

結果として, 予約スロットの重なりが発生した場合, それ以降の予約スロットの取り方としては, random モデルのように現在のフレーム状況に左右されず無作為に予約スロットを選択する方式が確率的によいが, 予約スロットの重なりが発生していない場合には, 選定モデルの方がより予約スロットの重なりが起きにくいことがわかった。

確認する隣接スロットの数だが, Spec2.1, Spec2.2 では 2 つ確認するよりも 1 つ確認する方が確率的に優位であるがその差はわずかであり, 加えて, Spec2.3 では 2 つ確認する方が大幅によい結果を出している。よって全体的に見て 1 つより 2 つを確認したほうが確率的によりよい結果を得ることができるとわかった。

結果として, 一番よい方式であるのは初期状態で予約スロットの重なりがないとするなら, それ以降予約の重なりが最も起こりにくく, たとえ重なりが起こったとしてもしばらく時間が経過すると再び予約が重ならず取ることのできる det2 の方式であるといえる。

6 おわりに

本論文では AIS で用いられている送信予約システムについて, 予約スロットの決め方を少しずつ変化させたモデルを作成し, PRISM を用いて確率の計算と解析, 検証を行った。

結果として, 現実の方法に最も近い prob1 では短い時間の間で予約スロットの重なる確率が 0.78 以上で, 連続でバースト状態が続く確率が 0.25 以下となった。この確率はかなり高めだが, 少し長い時間が経過するとそれぞれの確率は 0.99, 0.01 となり, そのうちバーストが起らず送信が正しく行われるようになる。

また, 本研究で行った解析および検証では prob1 よりも det2 の方がよりよい方式であると言え, 予約スロットのより良い決め方があることを示している。

今後は, 実際の AIS の予約スロットの決定方法に予約スロットが連続で重なる確率が低くなるような工夫を行い, どんな状態においても低い確率が得られるような方式を提案し, 定量的な解析, 検証を行いたい。

また、本論文では船舶の数が3、スロット数が10と15と規模が小さかったので、今後は状態数の削減を行い、船舶とスロットを増やし、解析と検証を進めたい。

参考文献

- [1] 塩地誠, 水城南海男, 矢内崇雄, 中島敏和, 小林健, 大塚賢, *AIS* 情報による海上交通管理システム高度化, 沖電気工業 管制システム部, 2001.
- [2] 海上保安庁灯台部 電波標識課信号施設室, *AIS (船舶自動識別システム) Universal ship-borne Automatic Identification System*, 1999.
- [3] B.Berard, M.Bidoit, A.Finkel, F.Laroussinie, A.Petit, L.Petrucci, P.Schnoebelen, P.McKenzie, *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer, 2001.
- [4] A.Bianco and L.de Alfaro, *Model checking of probabilistic and nondeterministic systems*, In P. Thiagarajan, editor, Proc. 15th Conference on Foundations of Software Technology and Theoretical Computer Science, volume 1026 of LNCS, pp499-513, Springer, 1995.
- [5] C.Baier, J.P.Katoen and H.Hermanns, *Approximate symbolic model checking of continuous-time Markov chains*, In J. Baeten and S. Mauw, editors, Proc. 10th International Conference on Concurrency Theory (CONCUR'99), volume 1664, pp146-161. Springer, 1999.
- [6] K.L.McMillan, *Symbolic Model Checking*, Kluwer, 1998.
- [7] T.Sekizawa, T.Tsuchiya, K.Takahashi and T.Kikuno, *Probabilistic Model Checking of the One-Dimensional Ising Model*, IEICE TRANS.-INF.SYST., vol.E92-D, pp1002-1011, 2009.
- [8] M.Kwiatkowska, G.Norman and D. Parke, *PRISM: Probabilistic Model Checking for Performance and Reliability Analysis*, ACM SIGMETRICS Performance Evaluation Review, volume 36(4), pp40-45, 2009.
- [9] J.Holzmann, *The Spin Model Checker*, Addison-Wesley Professional, 2003.
- [10] 中島震, *SPIN モデル検査 検証モデリング技法*, 近代科学社, 2008.
- [11] G.Larsen, F.Larsson, P.Pettersson and W.Yi, *Automated Analysis of an Audio Control Protocol Using Uppaal*, In Journal of Logic and Algebraic Programming, Volumes 52-53, pp163-181, July-August, 2002.
- [12] T.Herault, R.Lassaigne, F.Magniette and S.Peyronnet, *Approximate Probabilistic Model Checking*, In Proceedings of Fifth International VMCAI 04, pp 73-84 of LNCS:2937, January 2004.
- [13] H.Hermanns, J.Katoen, J.M.Kayser, M.Siegle, *ETMCC: Model Checking Performability Properties of Markov Chains*, Proceedings of the 2003 International Conference on Dependable Systems and Networks p673, 2003.
- [14] H.L.S. Younes, *Ymer: A Statistical Model Checker*, In Proceedings of the 17th International Conference on Computer Aided Verification, Volume 3576 of Lecture Notes in Computer Science, pp429-433, Springer. 2005.
- [15] 豊島崇士, *AIS* における送信予約のモデル化と検証, 関西学院大学理工学部卒業論文, 2008.