マルチメディア通信と分散処理 80－22
グループウェア 21－22
（1997. 1. 31）

分散協調型強化学習によるリフレクティブエージェントの性能評価

阿部 倫之　　中沢実　　服部 進実
金沢工業大学 情報工学科

Telescript などのモーバイルエージェントシステムは，ユーザの要求記述に基づいて広域ネットワーク上を移動しながら必要な行為を連続適用していくメカニズムを持っている．このエージェントの動作シナリオは，事前に全て記述する必要があるため，動的な環境変化を予測した記述をするのは困難である．本稿では，非決定的な動作シナリオをコンセプトと強度付きプロダクションルールで記述し，分散ルール強化学習メカニズムと分散ルール協調メカニズムによって動作シナリオを改変していくリフレクティブマルチエージェントシステム MAS/R を提案している．また，CLOS を用いて評価システムを実装し，MAS/R の適用能力を評価した．その結果，リフレクション（強化学習）を繰り返すことよって不適切なルールが淘汰され，環境変動に適応するように動作シナリオが改変されることを確認した．

Ability Evaluation of Reflective Agent Based on the Distributed and Cooperative Reinforcement Learning

Noriyuki ABE, Minoru NAKAZAWA, Shimmi HATTORI
Kanazawa Institute of Technology

Mobile agent system like Telescript in distributed computing environment has the mechanism in which required actions are applied one after another according to users described scenario, moving agents on wide area network. However, it seems to be difficult to respond to open and dynamic network environment, because action scenario of agent behavior has to include all description of it prior to execution of agents. In this paper, reflective multi-agent system MAS/R which can autonomously change scenario of behavior by mechanism of distributed rule reinforcement learning and cooperation, describing behavior scenario of agent by concepts and production rules with strength, is proposed. This system connected to multi-server load simulator has been implemented and evaluated. As the result of it, behavior scenario of agents has been confirmed to be autonomously changed to adopt to dynamic environment, selecting appropriate rules by repeating reflection.

1 はじめに

インターネットに代表される開放型広域分散ネットワークは，インターオペラビリティやスケーラビリティを重視したよりフレキシブルな分散コンピューティング環境の提供を可能にしている．Telescript[7] などのモーバイルエージェントシステムは，ユーザの要求記述に基づいて，広域ネットワーク上に存在しているサービス資源に移動しながらネゴシエーションを行ない，必要な行為を連続適用していくメカニズムを持っている．このエージェントの動作シナリオは，事前に全て記述する必要があるため，動的な環境変化を予測した記述をするのは困難といえる．このためには，環境変動に追従しながら適応的に動作シナリオを改変（学習）することで，リアルタイムに最適な行為を選択できることが重要と考える．このような機能を実現するマルチエージェントシステムでは，自己の行為の適用結果によってリフレクティブに行為（知識）の洗練や導出を行なう経験強化型の学習メカニズム [1, 6] が必要である．我々は，非決定的なエージェントの動作シナリオを，コンセプトと強度付きプロダクションルールで記述し，分散ルール強化学習メカニズムと分散ルール協調メカニズムによって動作シナリオを改変していくリフレクティブマルチエージェントシステムを提案している [9]．本稿では、このシステムの概要と評価結果について述べる．

2 分散協調型強化学習によるマルチエージェントモデル

2.1 マルチエージェントモデル

エージェントの動作シナリオはプロダクションルールによって記述する．このルールには，強度と支持度を表現した値を付加することができ，この値を付け値（bid）と呼ぶ．図1にマルチエージェントモデルの概要を示す．エージェント内での推論は，通常の認知一行動サイクルに基づいて，ルール条件部のマッチングと行為部の実行を目標に至るまで繰り返す．行為部の実行によってメッセージが発行され，エージェントの作業記憶（working memory）の

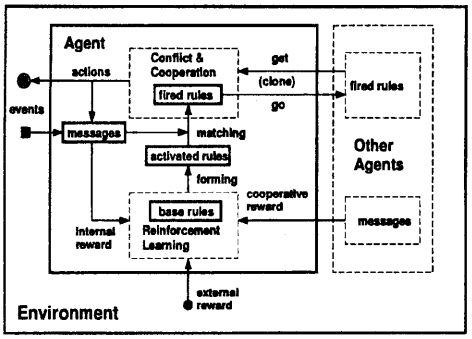

図 1: マルチエージェントモデル

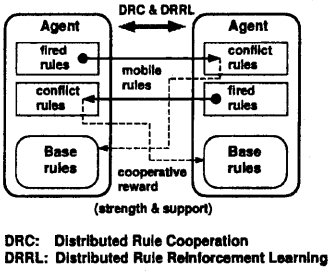

図 2: 分散協調モデル

内容が変化する．また，作業記憶は環境からのメッセージも格納するため環境変動により変化する．

支持度 (support) は，ルールの活性化状態を表現しており，設定した閾値以上の支持度を持つルールを活性化ルール（activated rule）として扱うことで，環境変動に応じたルールの活性化パターンを形成する．

ルールの強度は，環境からの報酬（reward）によって変化し，環境に対する行為の適用結果が有効である場合に増加し，無効である場合に減少させることで不用なルールを自然淘汰する．これをルール強化学習と呼ぶ．発火ルールの行為は並列実行可能であるが，相互排他行為（mutually exclusive action）を持つ発火ルールがある場合，この強度と支持度から算出される付け値を用いて競合解消を図る．これにより，ルール活性化パターンの中で環境に貢献してきたルールが発火し易くなり，結果として環境変動に適応的に自己洗練していくリフレクティブなメカニズムを実現できる．

2.2 エージェントの動作シナリオ

エージェントの動作シナリオは，ルール，目標，相互排他的カテゴリ（mutually exclusive category），学習ルールおよび作業記憶から成る．作業記憶内には，真理値を付加したメッセージが格納される．メッセージは，次の形式，

（コンセプト名（引数）真理値）

で表現する．ここで，コンセプトはオブジェクト指向におけるクラスに相当し，メッセージはコンセプトのインスタンスである．たとえば，host1 が host2 の server であることをメッセージ表現すると，

(server (host1 host2) true)

となる．作業記憶にメッセージが格納されるのは，発火ルールの行為部実行による場合と環境から直接与えられる場合とがある．ルールは，作業記憶の内容を判定する条件部とメッセージを発行する行為部から成る．ルール（行為部）は並列実行可能であるが，相互排他的カテゴリに属する行為（相互排他行為）は付け値を用いて競合解消し1つに絞り込む．ここで，環境に対して行為を適用した場合，適用後の環境変動状況と学習ルールとのマッチングが試みられ，報酬の決定とルール強化学習が実施される．

2.3 分散協調メカニズム

エージェント間の分散協調モデルを図2に示す．エージェントが協調するとき，互いに協調相手の発火ルールを自己の発火ルール集合に取り込み，付け値を用いて競合解消を図る。これを分散ルール協調（DRC:Distributed Rule Cooperation）と呼ぶ．ここで，発火ルールを送るエージェントを生産者エージェント，受け取るエージェントを消費者エージェントと呼ぶ．また移動する発火ルールをモーバイルルール（mobile rule）と呼ぶ．このモーバイルルールの実現は，動作シナリオの断片を取り込むことに相当しており，動作シナリオの改変を動的に実施するための共通メカニズムを与える．

消費者エージェントにおいて，モーバイルルールの行為部が実行された場合，その行為の適用結果を自分の学習ルールに基づいて評価し，報酬または罰則のメッセージを生産者エージェントに伝える．これを分散ルール強化学習（DRRL:distributed rule reinforcement learning）と呼ぶ．モーバイルルールが直接的協調であるのに対し，DRRL では，エージェント間において間接的で緩やかな協調を実現する．モーバイルルールは，相互排他行為を持つルールが発火した場合に限定して発生させることができるため，各エージェント内にある相互排他的カテゴリを調整することで協調のための通信量の制御が可能である．

2.4 分散ルール強化学習

分散ルール強化学習法は，profit sharing 法 [2, 1] と bucket brigade 法 [2]，さらに2つを併用した hybrid 法をベースにしている。ここで、分散 profit sharing 法と分散 bucket brigade 法による DRRL では，モーバイルルールによってエージェント間にまたがった分散強化学習を実施する．また hybrid 法では両者を併用した学習を実施する．

（1）分散 profit sharing 法

ルールの実行履歴を保持し，行為の適用後，学習ルールの判定によって与えられる報酬（罰則）を，関係するルール系列に対して均等分配する．モーバイルルールが実行されると，実行履歴がエージェント内に保持されるため，報酬（罰則）の分配対象となるルールにモーバイルルールが

含まれる可能性がある．このとき，モーバイルルールの生産者エージェントに対して報酬（罰則）メッセージを送り，該当するルールを強化する．

（2）分散 bucket brigade 法

メッセージを作業記憶に書き込むとき，発火ルールの強度をその付け値の量だけ減少させる．この発火ルールを生産者ルールと呼ぶ．次の推論サイクルにおいて，このメッセージによって発火したルールを消費者ルールと呼び，生産者ルールに対して自分の付け値の量を報酬として送る．このとき消費者ルールの強度は送った報酬の量だけ減じる．生産者ルールがモーバイルルールの場合，その生産者エージェントに対して報酬メッセージを送る．また消費者ルールがモーバイルルールの場合は，その生産者エージェントに対して消費者ルールの強度を減少させるためのメッセージを送る．ここで，エージェント内に取り込んだモーバイルルールが消費者ルールになる可能性があるのは，このルールが次の推論サイクルまでエージェント内に留まった場合であり，モーバイルルールの滞留時間に依存する．

（3）分散 hybrid 法

分散 profit sharing 法と分散 bucket brigade 法を併用する．併用法としては，同時適用と動的切替えの2種類がある．同時適用の場合，それぞれ得られた報酬の 50 %を学習に用いる．また動的切替えの場合，その切替えのタイミングとしては，環境から報酬が得られた時点で実施する方法と推論途中で実施する方法がある．

3 リフレクティブマルチエージェントシステム

DRRL と DRC を用いて自己改変能力を持つリフレクティブマルチエージェントシステム（MAS/R: Reflective Multi-agent System）を構成する．システムは CLOS (Common Lisp Object System) を用いて SUN ワークステーション（S-7/300）上に実装した．

3.1 リフレクション機構

自己反映計算可能なメカニズムを有するシステムをリフレクティブシステム（reflective system）と呼ぶ [5]．提案する MAS/R では，DRRL と DRC に基づくリフレクション，およびメタ推論に基づくリフレクションがある．

（1）DRRL と DRC に基づくリフレクション

ベースルールによって記述されるエージェントの振る舞いは，DRRL によるベースルールの淘汰状況に依存する．DRRL を制御している学習ルールは，ベースルールと同様のプロダクションルールによって記述されており，作業記憶の内容に基づいて強化学習が実施される．これにより，競合解消に勝利するルールが変化し，さらに活性化するルールが変化して，動作シナリオのリフレクションが進行する．

また，DRC では，モーバイルルールを取り込み，競合解消フェーズを経て実行することにより，動作シナリオのリフレクションが実現される．モーバイルルールが実行される可能性は，エージェント内の競合ルールの付け値に依存しており，またモーバイルルールの実行は，DRRL に影響を与える．

（2）メタ推論に基づくリフレクション

推論フェーズと強化学習モードを制御するために，メタ推論によるリフレクションを導入する．これは，DRRL

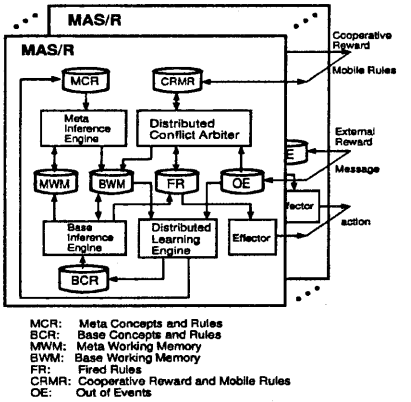

図 3: MAS/R のシステム構成

と DCR を含むベース推論過程を6つの処理フェーズに分割し，各処理フェーズをメタ推論によって決定する．さらに DRRL における学習モード（profit sharing 法，bucket brigade 法，hybrid 法）をメタ推論で決定可能する．従って，マルチエージェントシステムの場合，各エージェントは異なる学習モードで動作する可能性があり，このときモーバイルルールは消費者エージェントの学習モードに依存する．このメタ推論に基づくリフレクションでは，学習推論シナリオをメタルールで記述し，メタ作業記憶の内容に基づいて推論フェーズを決定する．これにより，エージェントの推論学習シナリオを動的に変更し，動作シナリオを間接的に改変するメカニズムを実現する．

3.2 システム構成

システム構成を図3に示す．システムは，ベース推論エンジン（base inference engine），メタ推論エンジン（meta inference engine），分散学習エンジン（Distributed learning engine），分散競合解消器（Distributed conflict arbiter），効果器（effector）から成る．

（1）メタ推論エンジン エージェントの状態と環境からのメッセージに基づいて，エージェントの次の処理フェーズと学習モードを決定する．学習モード変更の場合は，分散学習エンジンにメッセージを通知する．

（2）ベース推論エンジン メッセージの内容に基づいて，発火ルールを判別し、分散競合解消器，効果器，分散学習エンジンを呼び出す．

（3）分散競合解消器 相互排他行為を持つ発火ルールを検出する．このとき，協調領域からモーバイルルールを取り込み，発火ルールに含めて競合解消する．

（4）効果器 発火ルールの行為部と目標を照合し目標判定する．また，発火ルールの行為部を実行し，作業記憶（メタ作業記憶）の変更と環境への行為の適用を実施する．

（5）分散学習エンジン 学習モードの選択と報酬（罰則）の決定を行ない，ベースルールの強化学習を実施す

る．また，モーバイルルールを強化する場合は，その生産者エージェントに対して報酬（罰則）を送る。

4 評価

4.1 評価システム

MAS/Rにマルチサーバ負荷シミュレータを結合し，サーバ負荷平滑化制御を試みた．ここでは，動画のオンデマンドサービスを提供するマルチサーバ環境を想定する．エージェントは1つのサーバを管理し，必要に応じて他のエージェントと協調する．クライアント群からのサービス要求(request)がエージェントに通知されると，エージェント間で協調を行ない適切な処理サーバを決定し，クライアントとサーバをリンクする．リンクが確立されると動画の処理負荷を抽象化したトランザクションがサーバに送出される．エージェントはサーバの処理負荷を監視し，各サーバの処理負荷が平滑化するように協調を行ない，必要に応じて処理品質（QoS）の変更やサーバの切り替えを動的に実施する．このマルチサーバ負荷シミュレータをシミュレーション言語SLAMとFortranを用いて作成し，MAS/Rとリンクして連動する評価システムをS-7/300上に構築した．シミュレーションに用いたリクエストとトランザクションの定義を次に示す．

（1）リクエスト　クライアントは，処理負荷の異なる動画サービスを要求する．リクエストのパラメータは，サービスの種類，要求時間，サービス品質（QoS）である．サービス品質は，許容するトランザクション損失率で表現し，0％，20％，40％の3種類がある．

（2）トランザクション　動画デバイスからサーバに送られる動画フレームをトランザクションで表現する．デバイスからサーバに送出されるトランザクションは，サーバ内で待ち行列を形成し順次処理される．

トランザクションの単位は，30frame/secの動画を，15frame/GOP（group of picture）でMPEG圧縮したメディアを基準にする．そこで，トランザクションの単位をGOPとして考え，発生間隔は平均0.5秒、リクエストは平均30秒のランダム到着とした。また、サービス品質は、許容トランザクション損失率で表現した。

4.2 ベースルールとメタルールの構成

（1） ベースルールの構成

主なルールの種類は，行為決定ルール、サーバ切替え先決定ルール、サーバ品質決定ルール、サーバ接続ルールであり，70個のルールで動作シナリオを構成した．ここで，行為決定ルールには，無動作，サーバ切替え，サービス品質upとdownがあり，単位時間あたりのサーバ処理負荷に応じて，条件部のみ異なる複数のルールを設定した．サービス品質downルールの例を次に示す．

```
IF
((Event(=S-ID =X current-load)true)
 (Event(=S-ID level-2 current-QoS)true)
 (Eval((range(=X [1300 1500])true))true))
THEN
((Action
  (QoSUpdate =S-ID QoS-down ) true))
```

このルールはサービス品質のdownを行為として持つルールである．サーバ負荷が条件部のrange内でかつ現在の品質がlevel-2の場合に発火ルールとなる．この行為については、条件部のrangeが異なるルールを4つ定義しており、その内1つが発火ルールとなる。無動作ルールとサーバ切替えルールからもそれぞれ1つ発火するようにルール構成しているが，この3つの行為は相互排他的カテゴリとして設定しており，付け値による競合解消によって1つの発火ルールに絞り込まれる．

（2）メタルールの構成

メタルールでは，ベース推論のフェーズ決定と学習モードの決定を行なう．今回、ベース推論の処理フェーズの決定を固定アルゴリズム（決定的なルールで記述）とし，学習モードのみ動的切替えできるようにメタルールを設定した．学習モードの切替え基準としては，「正の報酬が得られた場合bucket brigade法，負の報酬（罰則の報酬）が得られた場合，profit sharing法を適用する」という単純な戦略を設定した．これは，不適切なルールを早期に淘汰する方向にエージェントを自己改変させることが狙いである．メタルールの例を次に示す．

```
IF
((MetaEvent(=X current-reward)true)
 (Eval(< =X 0 )true))
THEN
((MetaAction
  (LearningMode profit-sharing) true))
```

4.3 学習ルールの構成

行為適用後の環境変化と期待される効果との差によって報酬または罰則を学習ルールを用いて決定する．例えば，「ある行為によってサーバの負荷が設定値以上に上昇した場合，この行為を生成したルールに対して罰則を与える」ための学習ルールの例を次に示す．

```
IF
((Event(=S-ID =X recent-load)true)
 (Event(=S-ID =Y current-load)true)
 (Eval( > (- =Y =X) 1000 )true))
THEN
((Reward((Action
          (QoSUpdate =S-ID QoS-up)
          true) -10)true))
```

分散学習エンジンは，Rewardの引数にあるメッセージと旧ワーキングメモリ（環境に行為を適用した時点のワーキングメモリ）の内容とマッチングを試みる．マッチングに成功した場合，このメッセージを生成したルールの強度を学習モードに従って修正する．強度修正の対象となるルールがモーバイルルールの場合，その生産者エージェントに対して報酬（cooperative reward）を送る．DRRLの学習モードは各エージェントのメタルールにより決定されているため，協調報酬の値は，モーバイル先のエージェントの学習モードに依存する．

4.4 結果と検討

シミュレーションの1サイクル時間を3600秒として設定し，15秒間隔でSLAMの環境情報（トランザクション数，サーバ待ち行列長，リクエスト情報）をエージェント（MAS/R）に送り，エージェントからは推論遅延時間無しでSLAMに対し指示（サーバ切替え，サーバ接続，QoS変更）を送った．エージェントは15秒の割り込み間隔で環境情報を受け取り，サーバに対する指示（エージェントの目標）を導出しているため，シミュレーション1サイクル当り240回の指示と強化学習を時系列的に実施している．シミュレーションサイクルを継続する際は，MAS/R側の学習結果（強度と支持度）を保持した状態でSLAM側のみを初期化する．すなわち，シミュレーションサイクルを

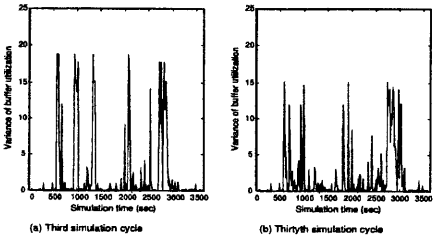

図 4: 学習能力の評価（profit sharing 法）

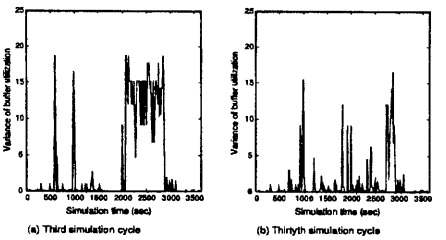

図 5: 学習能力の評価（bucket brigade 法）

繰り返すことにより，負荷変動パターンにマッチしたルールが早期に選択されるように自己改変していく能力を評価する．実験では，リフレクションの効果を確認するために，４つの学習モードで MAS/R を稼働させた．学習モードは、profit sharing 法 bucket brigade 法、固定 hybrid 法、動的 hybrid 法である。

固定 hybrid 法は，profit sharing 法と bucket brigade 法を同時に実行する学習モードである．また，動的 hybrid 法は，報酬が与えられた時点で profit sharing 法と bucket brigade 法をメタ推論によって相互に切替える学習モードであり，メタ推論に基づくリフレクションの効果を確認する目的で実施した．

まず，シミュレーション３サイクル目と３０サイクル目におけるサーバ負荷 (トランザクションの滞留個数) の分

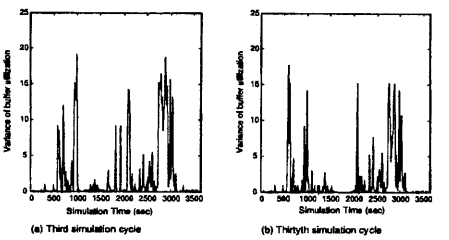

図 6: 学習能力の評価（固定 hybrid 法）

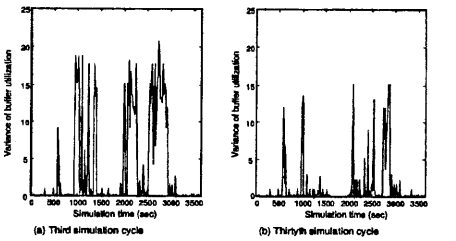

図 7: 学習能力の評価（動的 hybrid 法）

散値を各学習モード毎に計測したグラフを図４から図７に示す．横軸はシミュレーション時間を表しており，グラフの分散値が０に近いほど負荷が平滑化されていることを表している．このグラフより，全ての学習モードで３０サイクル目において分散値が減少方向に推移しており，DRRL と DRC に基づくリフレクションによって，負荷平滑化の方向にルールが絞り込まれ，シナリオが動的に改変されているのがわかる．ここで，図４の profit sharing 法と図５の bucket brigade 法を比較すると，シミュレーション初期段階（３サイクル目）においては，profit sharing 法に有利な結果が得られている．これは profit sharing 法が，不要ルールを淘汰するための学習速度が早いことによるものである．しかし，３０サイクル目においては，bucket brigade 法に有利な結果が得られている．これは，bucket bridage 法が過去の学習履歴を詳細に保持している期間が長いため，環境変動を予測したルールの絞り込みが効果的に実施されていることによるものと考える．

図６の固定 hybrid 法（fixed hybrid）では，profit sharing 法と bucket bridage 法の両者の利点が混在した中間的な結果が得られている．特に，profit sharing 法が bucket brigade に影響を与える形で学習が進行しており，動的 hybrid 法よりもシミュレーション初期段階では良い結果が得られている．また，図７の動的 hybrid 法（dynamic hybrid）では，メタ推論によるリフレクションによって profit sharing 法と bucket brigade 法を動的に切替えており，３０サイクル目においては図５の bucket brigade 法と同様の結果が得られている．この学習モードでは，bucket brigade 法を主体にして profit sharing 法を断片的に適用しているため，bucket brigade 法の利点を保持しながら profit sharing 法の利点を取り込んでいる．

次に，１シミュレーションサイクルにおける分散値の平均値を算出し，シミュレーションサイクルの継続に伴う学習収束状況を表したグラフを図８に示す．このグラフにおいて，最小値を得ているのは bucket brigade 法であるが，２５サイクルから３０サイクルのシミュレーション後半において再び大きく振動している．また，安定しながら収束方向にあるのは動的 hybrid 法であることがわかる．動的 hybrid 法は，５サイクル付近のシミュレーション初期段階においても良い結果を得ており，メタ推論に基づくリフレクション効果が顕著に現れている．固定 hybrid 法は profit sharing 法の影響が強いが，これが bucket brigade 法の振動を打ち消す働きをしており，シミュレーション後半において安定した結果を得ている．以上の結果，総合的に評価すると，動的 hybrid 法が最も環境変動に対する適

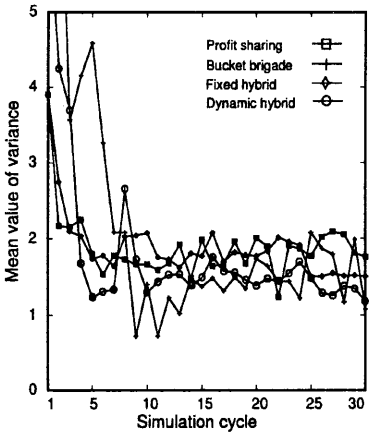

図 8: 学習シミュレーションサイクルの収束

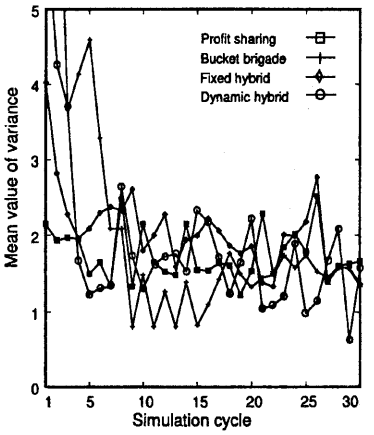

図 9: 学習シミュレーションサイクルの収束（推論遅延モード）

応性が高いものと考える．

ここで，エージェントの推論遅延を考慮した場合の学習収束状況を図 9 に示す．各エージェントの推論遅延時間は 10 秒に設定した．この遅延時間は平均トランザクション到着間隔の 20 倍である．この場合，エージェントの行為は 10 秒先の環境状態に適用されるため，エージェントとしては環境変動を予測した行為を適用するようにシナリオを改変していく必要がある．図 9 のグラフでは，全ての学習モードにおいて図 8 よりも負荷変動が大きく現れている．ここで動的 hybrid 法については，シミュレーション後半において分散値が減少する方向に収束しており，他の学習モードに比べて良い結果を得ている．また，最小値として見ると図 8 と比べて同等の結果を得ているため，先読み行為の適用によって見かけ上推論遅延時間が吸収可能であることがわかる．特に動的 hybrid 法については，メタ推論による学習モードの切替え戦略をさらに洗練することで環境適応能力を向上させることが可能と考える．

5 おわりに

本論文では，分散協調型強化学習によるリフレクティブマルチエージェントシステム MAS/R について述べた．また，MAS/R のリフレクション（学習）能力とその効果を確認するために，動画のマルチサーバシミュレータと結合して評価システムを作成し適応性を評価した．その結果，動作シナリオが環境変動に適応するように改変され，不要なシナリオが自律的に淘汰される方向に収束することを確認した．ここで、本システムの環境変動への適応能力は，シナリオを記述しているルールの粒度に依存している．しかし，ルールの強度が粗いルール群を補間する可能性があるため，今後，ルールの粒度と強度に関して詳細に検討していく必要があるものと考える．

謝辞

本研究は（財）テレコム先端技術研究支援センター殿の助成研究の一環として行なわれたものである．

参考文献

[1] Grefenstette,J.J.,"Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms", Machine Learning,Vol.3,pp.225-245(1988)

[2] J.H. ホランド, K.J. ホリオーク, R.E. ニスベット,P.R. サガード著，市川伸一ほか訳,"インダクション INDUCTION",p.462, 新曜社 (1991)

[3] Tan,M.,"Multi-agent Reinforcement Learning : Independent vs. Cooperative Agents", Proc.10th Int. Conf. on Machine Learning,pp.330-337(1993)

[4] WeiB,G.,"Learning Coordinate Actions in Multi-Agent Systems", Proc.13th Int. Joint Conf. on Artificial Intelligence,pp.311-316(1993)

[5] 渡辺卓雄,"リフレクション"，コンピュータソフトウェア，Vol.11,No.3,pp.5-14, 日本ソフトウェア科学会 (1994)

[6] 畝見達夫,"強化学習", 人工知能誌, vol.9,no.6,pp830-836(1994)

[7] James E.White,"Telescript Technology: The foundation for the Electronic Marketplace",General Magic White Paper,p.24,General Magic,Inc.(1994)

[8] 阿部倫之，中沢実，服部進実,"コンセプトネットワークによるルール強化学習に基づくマルチエージェントシステム" 信学論 B-I,Vol.J79-B-I,No.5,pp.226-237(1996)

[9] 中沢実, 阿部倫之, 服部進実,"リフレクティブエージェントコンピューティングによるネットワーク制御方式" マルチメディア通信と分散処理ワークショップ論文集, Vol.96,pp.1-8(1996)