

分散協調PIによるマルチエージェントシステム

3J-9

目黒雄峰 井田通 阿部倫之 服部進実

金沢工業大学 人間・情報・経営系

1 はじめに

ネットワークの負荷制御系（マルチメディアトラフィック制御、ルーティング制御等）では、通信リンクに流れるトラフィック量に応じて、ある戦略を選択する基準値（スレシヨールド値、QoS値）が動的に変化する。従って、ある特定の静的環境に対応した特定の戦略やサービスを選択するのは異なり、時間的に変動する連続量を制御対象とする必要がある。

このような動的負荷環境に適応可能な自律分散学習モデルとしては、連続量をリアルタイムで処理しうる非記号処理ベースの構成が要求される。ここで、記号処理/非記号処理を融合させたモデルとして、プロダクションルール荷重をバケツリレーアルゴリズムによりエージェント間で並列処理を行いながら修正することで概念を形成する（PIベースの）分散協調型帰納推論システム⁽²⁾がある。このモデルは、ルールベースシステムにニューラルネット等の非記号処理モデルにおける荷重修正学習メカニズムを組み込んだものであり、これに基づくエージェント間の通信粒度はルールおよび荷重である。

本稿では、連続量的環境変動に対応できるプラットフォームとして、Process of Induction⁽³⁾ (PI) の手法を分散環境に拡張した分散協調PIを示し、これによるマルチエージェントシステムの構成について述べる。

2 分散協調PI

2.1 PI

PIでは、ルールベースシステムにおける①「条件・行為ルール」とオブジェクト指向においてクラスに相当するフレーム型のデータ構造②「コンセプト (concept)」を用いて環境内の各種情報および関係のモデル化を行う。各ルールは何らかのコンセプトに所属しており、例えば Human というコンセプトには人間に関する事を条件部に持つルールが所属している。ここで、Human に所属するルールの行為部にコンセプト Die に関する事が記述されている場合（例えば「Xが人間であるならばXは死ぬ」など）、コンセプト Human と Die はそのルールによって関係付けられているという（図1）。

PIはメッセージリスト（ルールベースシステムにおけるワーキングメモリ）と呼ばれる環境情報（メッセージと呼ぶ）とルールの条件部をマッチングし発火させる。発火したルール群は、並列にその行為部が実行される。但し、相互排他的行為を行うルールについては、付け値 (bit) と呼ばれるルールの活性度に基づいて競合を解消する。行為部の実行にはメッセージの送信も含まれており、これによってメッセージリストが更新され、他のルール集合が次のサイクルにおいて発火する可能性を持つ。ルールの付け値は、

強度 (strength)

過去にどの程度ルールが有効であったかを示す。目標達成に貢献するルールに連なっている場合、強化学習プログラムによって増加の方向に変化してする。

支持度 (support)

ルールが所属しているコンセプトの活性度がルールの支持度となる。

貢献度 (contribution)

ルールの行為部が目標や副目標をどの程度満足させるかを示す。

や特殊性（条件部の複雑さ）などのから算出する。

コンセプトには活性度 (activity) が付加されており、ある閾値に達するとコンセプトは活性化される。活性度は、発火しているルールの行為部でコンセプトが使われている場合に増加する。コンセプトの活性化、ルールのマッチング、行為部実行の1サイクルにおいて使用されなかったコンセプトは、その活性度が減少される。活性度がある閾値以下になるとコンセプトは非活性化し、これによって所属するルールもマッチングの対象から除外される。このサイクルを繰り返していく過程において問題に適応したコンセプト空間が組織化されていく。また、提案する分散協調PIでは、実行履歴を保持しないルールの強化学習方法として、分類子システムのバケツリレーアルゴリズム (bucket brigade algorithm)⁽⁴⁾ を採用している。

```
IF (Human (-X) true) THEN (Die (-X) true)
IF (Human (-X) true) THEN (Talk (-X) true)
```

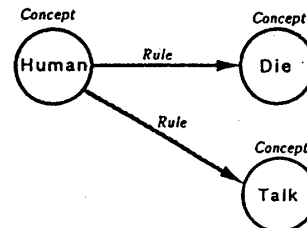


図1 ルールによるコンセプト間結合

Multi-Agent Model with Distributed Cooperative PI
 Yuuhou Meguro, Tooru Ida, Noriyuki Abe, Shimmi Hattori
 Kanazawa Institute of Technology
 7-1 Ougigaoka, Nonoi-chi-machi, Ishikawa-gun,
 Ishikawa, 921 Japan

2.2 分散協調PI

PIは、コンセプト、ルール、メッセージリスト（ワーキングメモリ）から構成されている。これをPIモジュールとして考え、複数のPIモジュールを分散環境下に配置すると、コンセプトはモジュール間にまたがるルールからも支持されることになり、この支持によって分散PI間の間接的な協調が実施される（図2）。

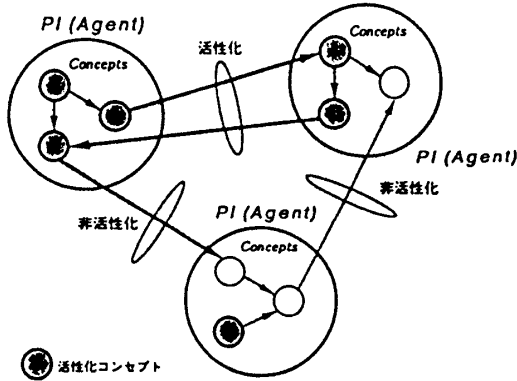
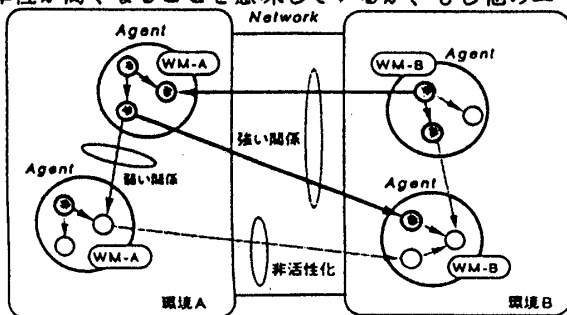


図2 分散協調PI

ここで、メッセージリストの内容が各PIモジュールで異なるため、相互排他的カテゴリ（同時に実行できない行為）に属するメッセージが発生した場合、他のPIモジュールの発火ルール群を自モジュール内に取り込み、分散PI全体としての競合の解消を図る必要がある。これは直接的な協調といえる。バケツリレーアルゴリズムはモジュール間にも適用し、モジュール間にまたがった強度の伝搬を行う。このような協調動作を伴ったPIを分散協調PIと呼ぶ。この分散協調PIを用いて連続量的学習能力を持つマルチエージェントモデルを構成する。

3 マルチエージェントシステム

エージェントをPIモジュールで表現することにより分散協調PIはマルチエージェントシステムとなる。ここで、システムのエージェントがネットワークを介して協調する場合、ルール実行の1サイクル毎にネットワークにメッセージが送出されるため、オーバーヘッドを考慮するとある程度スループットを意識した協調動作が必要となる。そこで、協調動作は、エージェント内において相互排他的カテゴリに属するメッセージが生成された場合に限定するものとした。これは、エージェントの自律性が高くなることを意味しているが、もし他のエー



WM: Working Memory (message list)

図3 マルチエージェントモデル

ントから取り込んだ発火ルールが自エージェントにおいて発火した場合、コンセプトの活性度が増加して強度が相手に伝搬するため、結果として協調が行われたことになる。これは、先に述べた間接的な協調を間引きながらバッチ的に実施したのと同じ意味を持つ。

エージェントが協調する際の発火ルールの要求は、初期段階においてはブロードキャストによってなされる。このとき、取り込んだ発火ルールが自エージェントで発火した場合、その生産者エージェントとのルール強度が増加し関係が強化（活性化）される。逆に全く発火しなかった場合は強度が減少し関係が抑制（非活性化）される（図3）。したがって、関係が活性化している（関係の強度がある閾値以上）エージェントのみに発火ルールを要求することで、協調に伴うエージェント間の通信量を削減することが期待できる。

ここでエージェントが、協調すべきエージェントの環境に直接移動することで協調に伴うスループットの向上を図るエージェントマイグレーションの考え方がある。このマイグレーションは、事前に設定したシナリオに基づいて意識的に実施する方法が一般的である。しかし、時々刻々と変化する動的環境において、考えうる最適なシナリオを事前に設定するのは困難といえる。

このマイグレーションを分散協調PIに自然な形で取り込む方法として、エージェント間の活性度を評価して適応的にマイグレーションさせる自律的マイグレーションの実現を考えている。各エージェント自身は、マイグレーションのための明示的なシナリオを事前に持たず、マイグレーション先の候補が（関係の活性化の程度から）洗い出されると、その必要性の可否が閾値によって自律的に判定される。これにより、マイグレーションのシナリオは、変動するコンセプト間の活性度に基づいて適応的に組み立てられる。

4 おわりに

本稿では、連続量を対象とした自律分散学習モデルとして、PIを分散環境に拡張した分散協調PIを示し、これによるマルチエージェントモデルについて述べた。現在、分散協調PIを汎用マルチエージェントプラットフォームとして実装中であり、そのサブセットを用いてネットワークにおける動的ルーティングやサーバ負荷制御等への適応性について評価を進めている。

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われたものである。

参考文献

- (1) 山田誠二: "リアクティブプラン", 人工知能学会誌, Vol.8, No.6, pp.35-41, 1993
- (2) 服部進実: "分散協調型帰納推論エージェントによるサーバ負荷分散処理", 情報処理学会, マルチメディア通信と分散処理, 68-10, 1995
- (3) J.H.Holland et al.: "Induction: Process of Inference, Learning and Discovery, the MIT press, 1986