

自律分散学習モデルによるネットワークの動的負荷制御

服部進実

阿部倫之

金沢工業大学 人間・情報・経営系

〒921 石川県金沢南局区内野々市町扇が丘 7-1

E-mail: hattori@infor.kanazawa-it.ac.jp

分散システム環境で動的に変化する通信網の負荷に対し、メディア処理やルーティング処理を行う場合に、状況に応じた最適戦略を学習により自動的にかつ実時間で獲得する方法が要求される。本稿では、かようなリアルタイムかつトラヒック量等の連続量を対象とした環境に適応しうる自律分散学習モデルとして分散協調型 Process of Induction (PI) を取り上げ、その適用性を考察する。特に分散エージェント数が多い場合の協調作業量の増大はリアルタイム性に矛盾するため、通信網の負荷等の連続量に対し、処理遅延や転送遅延を見越した負荷量予測を行いながら適切な戦略を学習により自動的に獲得する仕組みが重要であることを示す。

Dynamic Load Control in Network by Distributed and Autonomous Learning Model

Shimmi HATTORI

Noriyuki ABE

Faculty of Engineering

kanazawa Institute of Technology

7-1, Ougigaoka, Nonoichi-machi, Kanazawa-south area, Ishikawa, 921 Japan

E-mail: hattori@infor.kanazawa-it.ac.jp

The optimized strategies or schedulings acquired by automatic learning are required to react to dynamically changed load of network in distributed system environment. In this paper, we consider the ability of applying the distributed and cooperative process of induction model for reactive and continuous environment. And also, an automatic acquisition of optimized strategies predicting the load covered with processing and transmitting delay is important, in order to compensate decrease of reactivity due to cooperative job among increasing numbers of agent.

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ヒック制御、ルーティング制御等）の世界で自律エージェントを取り巻く環境は、時々刻々変化する動的な世界である。かような動的な世界に対応する実時間プランニングの研究は、リアクティブプランニングとして一部で研究が行われている⁽¹⁾⁽²⁾。

かようなネットワーク負荷制御系に対応しうる自律エージェントに要求される条件としては、下記が考えられる。

(1) 環境の変動は連続量である。

例えば分散型ルーティングアルゴリズムとして波及型アルゴリズム等が提案されている⁽³⁾が、局所判断に基づいて目標ノードに到達する経路を求める為、最適経路の近似解を求めるものとなっている。通信ノードとリンクが複数のエージェントにより分散管理されているといった分散型経路探索問題では、リンクに流れるトラヒック量に応じて通信遅延はもとより、ある戦略を選択する基準値（スレシールド値、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。

(2) 環境は動的に変化し、実時間的対応が要求される。

通信網のノードにランダムに到着するコールは、発信地、目的地、通信開始時間、通信継続時間、通信バンド幅、優先度等の属性により記述される。このコールに対し、分散協調アルゴリズムによりルーティングスケジュールを行うには、コール群 $x(t)$ に対する可能スケジュール u のコスト関数 $C(x(t), u)$ を最小にするスケジュールを選択する必要がある。しかし問題は、自律エージェントによる分散協調作業により $x(t)$ を割り付けるのに T 時間を必要とすると、この間にコールは $x(t)$ から $x(t+T)$ へ移り $x(t+T)$ においては必ずしも最適解とならない。したがって、協調作業を行う自律エージェント数が多い場合には、処理遅延、通信遅延が増大し、システムが連続時間で変化する状況に対し実時間的対応が困難となる。かような環境変動に対する即応性と最適性のトレードオフ問題については、自律ロボット等の分野で検討が始まったばかりであり、未だ汎用プラットフォームとなりうるモデルは発表されていない。

本稿では、自律エージェント間の通信粒度により自律分散学習モデルを非記号処理モデル(ニューラルネット)と記号処理モデル(マルチエージェント)に分類し、J.H.Hollandらが提唱する Process of Induction⁽⁵⁾ (PI)の手法を分散環境に適応しうるモデルに拡張展開したものを上記モデルの中間に位置付けた。さらに、前述の通信網の実時間かつ連続量の動的環境に適応させるには、並列実行処理、予測学習機能をモデルに含ませることが重要であることを述べる。

2. 自律分散学習モデル 一 記号/非記号融合処理モデル

計算モデルとしての自律分散学習モデルについては汎用的なプラットフォームとして種々の応用に耐え得るレベルのものはまだ発表されていない。

図1は自律分散学習モデルについて、自律的な動作主体(これをエージェントとする)の粒度(単純なエージェント/複雑なエージェント、エージェント間通信単位等の観点)により分類したものである。粒度の小さいものはニューラルネットワークにみられる如く非記号処理モデルであり、エージェント間の通信量は誤差量又は荷重修正量であり、学習のさせ方に依存するが連続量の環境にも適応させうる。

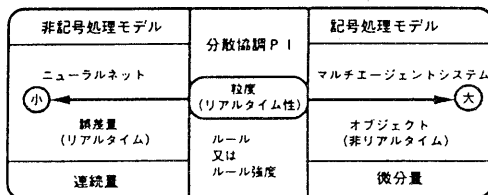


図1 自律分散学習モデル

一方、粒度の大きいものは、分散オブジェクト指向計算モデルに代表されるマルチエージェントシステム等の記号処理モデルであり、エージェント間の通信は、エージェント間の協調や相互作用の枠組みを規定するプロトコルを含むオブジェクト単位として行われる。適用される領域としては、オブジェクトやエージェントの抽象化機能、さらには推論機構により、環境に適応した戦略やサービスの特定を行う離散量、又は微分量の世界である。このことは、エージェントの機能が高度化、複雑化し、協調エージェントの数が大きくなる程協調作業量が増大し、リアルタイム環境には不適格であることを意味する。

通信網の動的負荷環境に適応しうる自律分散学習モデルとしては、前述の如く連続量をリアルタイムで処理しうる構成が要求されるが、記号処理/非記号処理を融合させるモデルとして、後述するように、プロダクションルール荷重をパ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分散協調型帰納推論システム⁽⁴⁾が考えられる。このモデルの原形は、遺伝アルゴリズムやニューラルネット等の非記号処理モデルの荷重修正に基づくものであるが、エージェント間の通信粒度はルールである⁽⁴⁾。

表1 各モデルの特徴

	ニューラルネット	分散協調 P I	エージェントシステム
基本アルゴリズム	誤差量修正	パケツリレーによるルール修正	オブジェクトマイグレーション
通信の粒度	荷重修正量(小)	ルール(中)	オブジェクト(大)
ドメイン依存性	小	中	大
自己組織化	学習フェーズ	ルールとコンセプトの光線と発現	エージェントの階層化
実時間性	良	可	困難
並列性	良	可	可
予測性	良	可	困難

表1に示す如く、分散協調 P I はドメイン依存性は比較的小さく、汎用プラットフォームを構築し得る可能性があり、実時間性、並列性も比較的实现させやすい。すなわち、動的環境に対する実時間性を向上させる方法として、時間方向の連続的環境変動に対しエージェントに予測学習機能を持たせることにより、処理遅延、通信遅延を見越した時点での適応解を探索することにより、見かけ上吸収することが可能となる。例えば、ニューラルネットワークにおいては、学習フェーズにて時間軸に対する経験的負荷変動を学習させることにより予測機能を持たせることが可能であることは周知の通りである。

一方分散協調 P I においてもルールベースに経験的知識(概念+ルール)×荷重を埋め込むことにより予測学習を行うことは可能である。

図2はネットワーク負荷スケジュールサーバの負荷状況を時刻 t_1 でサンプリング検知したエージェントが、エージェント間通信によるネゴシエーションを含めてコンセプトを獲得し、最適解をサーバに提示する(時刻 t_4) 状況を示したものである。ここで、エージェントがサーバの負荷状況を検知した時刻 t_1 で時刻 t_4 のサー

バ負荷を経験的知識より予測し、この時点での対応解を求めればよい。つまりエージェントの協調処理遅延、通信遅延 $\Delta(t_4, t_1)$ は吸収できることになる。以下にこれを実現するための分散協調PIの構成について述べる。

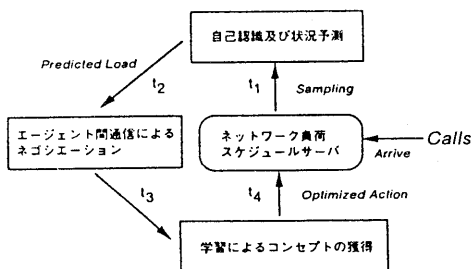


図2 ネットワークの動的負荷制御系

3 分散協調PI

3.1 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3)。コンセプトとルールの構造を図4に示す。

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

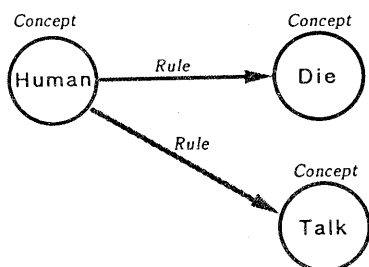


図3 ルールによるコンセプト間結合

コンセプトの構造

```
(defstruct concept
  (name nil) ; コンセプト名
  (arity 1 : type fixnum) ; 引数の個数
  (activity 1.0 : type rational) ; 活性度
  (super-concept '() : type (satisfies listp)) ; 上位概念
  (instance '() : type (satisfies listp)) ; 事例
  (rule-list nil : type (satisfies listp)) ; ルールリスト
  (environment '() : type (satisfies listp)) ; 環境
```

ルールの構造

```
(defstruct rule
  (name '() : type (satisfies listp)) ; コンセプト名
  (condition '() : type (satisfies listp)) ; 条件部
  (action '() : type (satisfies listp)) ; 行為部
  (strength 5.0 : type rational) ; 強度
  (contribution 1.0 : type rational) ; 貢献度
  (environment '() : type (satisfies listp)) ; 環境
```

図4 コンセプトとルールの構造

メッセージの基本形式

([述語 (コンセプト名)] [引数リスト] [真理値])

事例

(Human (服部) true)

メッセージリストの構造

(メッセージ メッセージ ...)

図5 メッセージとメッセージリスト

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、図5)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信頼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、ルールに付加されている強度(strength)、支持度(support)、貢献度(contribution)から算出する。

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、後で述べるバケツリレーアルゴリズムなどの強化学習プログラムによって増加の方向にリアルタイムに変化していく。

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。ルールの条件部が複数の場合、ルールは複数のコンセプトに所属しているため、支持度は所属しているコンセプトの活性度の和で表現する。

貢献度 (contribution)

ルール行為部が目標や副目標をどの程度満足させるかを示す。活性化（または発火）したルールの行為部と目標（または副目標）のマッチングが成功した場合、ルールの貢献度が増加する。

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する。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。このときコンセプトはそのルールから支持され、コンセプト内のルールも同様に支持される（そこで、コンセプトの活性度を所属しているルールの支持度としている）。

PIでは、活性化コンセプトに所属しているルールのみがメッセージリストとのマッチングの対象となるため、全てのルールが対象となるルールベースシステムとは異なる。ここで、最初に活性化しているコンセプトは、与えられた問題の初期条件と目標に関連するコンセプトである。活性化の拡散は、発火ルールが送出すメッセージによってコンセプトの活性化が伝搬していく前向き連鎖によるものと、目標と活性化ルールの行為部をマッチングしてそのルールの条件部を副目標とし、副目標に関連するコンセプトを活性化していく後向き連鎖によるものがある。

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。

PIにおける帰納学習の中心は、①「ルールとコンセプトの洗練（強度、活性度の修正）」と②「ルールの条件部の一般化や特殊化による新ルールの生成および新概念（コンセプト）の生成」である。特に②は、記号処理による形式的アプローチとなるため、条件部や行為部の形式的パターンによる記号の変換や連結などを必要とする。また無秩序な生成によるコストの増加を抑制するために起動条件を明確に定める必要がある。

ここで、PIとコネクションistモデルを対比させた場合、コンセプトがノード（要素単位）、ルールがコネクション（結合）に相当する。コンセプトの活性度をノードの入力値に、ルールの強度をコネクションの荷重に置き換えた場合、コネクションistモデルの荷重強化を中心にした（非記号処理による）連続量的な帰納学習による予測能力をPIに自然に取り込むことが期待できる。ここで、実時間的な環境変動に対して柔軟追従し、予測能力を維持していく上でポイントとなるのがルール強度の修正方法である。この分散協調PIでは、分類システムのバケツリレーアルゴリズム (bucket brigade algorithm) を採用しており次節でその概要を述べる。

3. 2 バケツリレーアルゴリズム

バケツリレーアルゴリズムは、条件部を満足するメッセージを生成するルール（生産者ルール）と生成されたメッセージによって条件部が満足するルール（消費者ルール）の関係をのみを扱う局所的な方法である。ある発火ルールの行為部が実行され、そのメッセージによってあるルールが発火した場合、消費者ルールは生産者ルールに対して自分の強度の一部（付け値）を支払う。生産者ルールはそれを報酬として受け取り自分の強度に追加する。消費者ルールの強度はこの時点において減少するが自分の行為部が生成するメッセージによって他のルールを発火できれば支払った以上の報酬を受け取る可能性を持つことになる。他のルールを発火させることができなければ強度は減少したままであり、このルールを含む系列が次のサイクルで再試行されていくと強度の減少が後ろ向きに伝搬してこの系列全体の強度が減少していく。逆に、ルールの発火が継続して目標に至ると生産者ルールに対して大きな報酬の支払いがなされ、その系列に連なるルールの再試行の度に強度の増加が後ろ向きに伝搬して系列全体の強度が増加していく。このイメージを図6に示す。

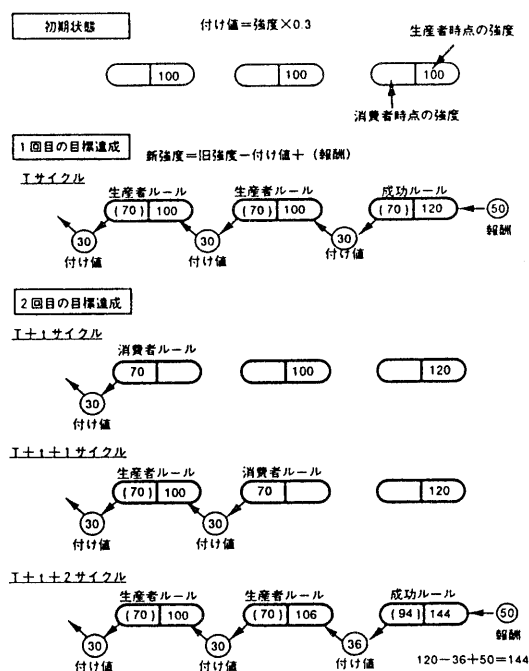


図6 バケツリレーアルゴリズム

強度伝搬による間接的なルール系列の強度の増減は緩やかに進行するが、目標に到達した時点でメッセージリスト内に留まっているメッセージ（メッセージリスト内のメッセージは相互排他的メッセージが到着するまでリスト内に留まっている）の生産者ルールに対して大きな報酬の支払いがなされるため、系列内に点在する幾つかのルールの強度は直接的に増加される可能性を持つ。この強度修正による帰納学習では、実行の履歴を保持しないでリアルタイムに行うことが可能であり、コネクショニストモデルにおける荷重の強化学習を実行時にインクリメンタルに行っていることに相当する。

3. 3 分散協調PIとエージェント

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った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7）。

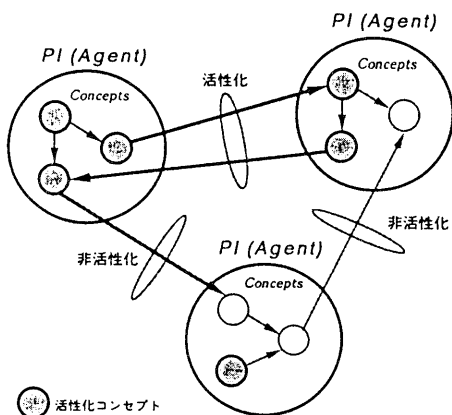


図7 分散協調PIとマルチエージェントモデル

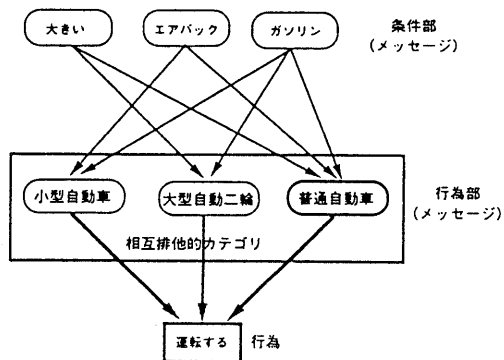


図8 相互排他的カテゴリ

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に属するメッセージ（図8）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。

パケツリリ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と呼ぶことにする。この分散協調PIを用いて連続量の学習能力を持つマルチエージェントモデルを構成する。

分散協調PIを直接適用した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、もし他のエージェ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の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9）。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。

また分散協調PIでは、現在必要としている関係がリアルタイムかつ自然に強化されるため、最適解を得るための自己組織化が、環境変動に対して適応的にスムーズに実施される。

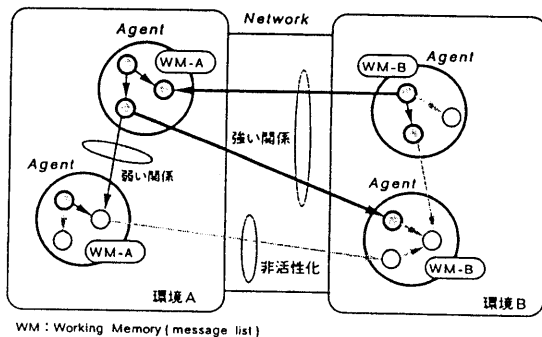


図9 自律分散エージェントモデル

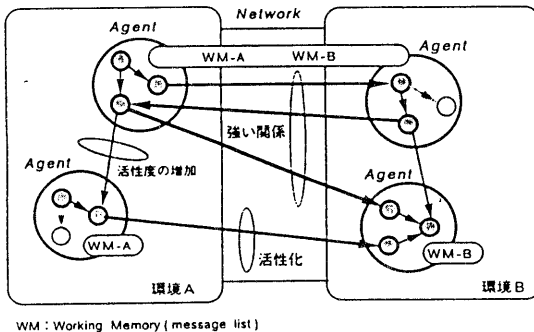
4 自律分散エージェントモデル

章3で述べたエージェントがネットワーク上に分散配置された場合、個々のエージェントおよびエージェント全体の自己組織化（学習）が適応的になされる。

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。

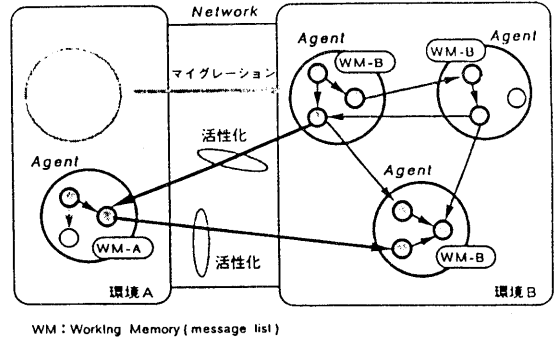
このマイグレーションを分散協調PIに自然な形で取り込む方法として、エージェント間の活性度を評価して適応的にマイグレーションさせる自律的マイグレーションの実現を考えている。これは、エージェント間の活性度が、通信の必要性和その効果に深く関係していることによるものである。各エージェント自身は、マイグレーションのための明示的なシナリオを事前に持たず、マイグレーション先の候補が（関係の活性化の程度から）洗い出されると、その必要性の可否を（閾値を用いて）自律的に判定する。これにより、マイグレーションのシナリオは、学習によって変動する活性度に基づいて適応的に組み立てられて行くことになる。

ここで、協調動作はスループットを意識して相互排他的メッセージの発生で起動する直接的協調のみに限定しているが、この場合、前章で述べたように相手のエージェントからの支持が間引かれて結果に反映されている。この影響によるルールの発火取りこぼしによって、活性度や強度に大きな誤差が生じているとマイグレーションの効果が低下する可能性がある。従って、これをできるだけ回避するために、エージェント間で一時的にワーキングメモリを共有する段階を作り（図10）、活性度の再評価を経てからマイグレーションを実施する（図11）方法を検討している。これは、マイグレーションを終了して元の環境に戻る場合も同様である。



WM : Working Memory (message list)

図10 自律分散エージェントモデルにおけるワーキングメモリの共有



WM : Working Memory (message list)

図11 自律分散エージェントモデルにおけるエージェントマイグレーション

5 おわりに

本稿では、通信網のリアルタイムかつトラフィック量等の連続量を対象とした環境に適応しうる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、これによる自律分散エージェントモデルについて述べた。

このモデルはドメイン依存性が低いため、エージェントマイグレーションまでも含めて汎用プラットフォームとして構成することが可能と考える。

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) 月岡陽一他: "状況に応じた戦略選択による実時間プランニング", 情報処理学会, 人工知能研究会99-20, 1995
- (3) 北村泰彦他: "波及型探索における通信制御法とその評価", マルチエージェントと協調計算Ⅲ, pp.185-192
- (4) 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
- (5) J.H.Holland et al.: "Induction: Process of Inference, Learning and Discovery, the MIT press, 1986