2004－ICS－136　(27)
2004／8／6

マルチエージェント信頼度割当て問題に対する報酬交換アプローチ

齋藤宗孝†　　大山義仁‡

北海道大学　電子科学研究所　情報数理研究分野

マルチエージェントシステムにおいてタスクが達成されるためには、役割分担が重要である。これをエージェントの学習によって実現するためには、各エージェントに適切に報酬が分配されなければならない。本稿ではタスクを達成するためのある段階を担うエージェントが、自身のひとつ前の段階の役割を担うエージェントに報酬を支払う枠組みについて考える。この枠組みに「交渉」を導入することが有益であるか否かを、計算機実験により検討する。その結果、エージェントが役割を果たすために多くのコストを支払わなくてはならないような場合、「交渉」を含むルールを用いたほうが、含まないルールを用いた場合よりも、システム全体としてのタスクの達成度が大きいことがわかった。これにより、エージェント間の報酬分配における「交渉」の有用性が示唆された。

Backward Reward Distribution for Multi-Agent Credit Assignment Problem

Munetaka Saitoh†　　Yoshihito Oyama‡

Nonlinear Studies and Computation, Research Institute for Electorical Science, Hokkaido Univ.

In multi-agent systems, role assignment is important for task achievement. For realizing role assignment by autonomous learning, reward must be suitably distributed to each agent. This paper considers ”backward reward distribution” framework. In this framework, agent contributing to some phase for task achievement pays reward to other agent contributing to one-step precedent phase. Whether ”negotiation” is useful or not in this framework is investigated by computer simulation. It is demonstrated that negotiation improves system’s overall performance under the conditions that agent must pay many costs for performing his role. Therefore, usefulness of negotiation in backward reward distribution is suggested.

1　はじめに

1.1　役割分担と報酬分配学習

マルチエージェントシステムがタスクを達成するためには、エージェントは協調して動作することが必要であり、そのひとつにエージェント間の役割分担が挙げられる。本稿では、タスク達成のために役割分担を実現する方法について考える。

役割分担の本質とは何であろうか。人間社会における一例として、「パンをつくる」というタスクを考えると、これは「小麦を生産する」というタスク、「小麦を粉にする」というタスク、「粉をこねてパンを焼く」というタスクなど、多くの役割から成る。１人でそれら全ての役割ができるようになるのは大変であるが、役割分担をすれば、各人が学習しなければならない技能の量は少なくなる。

エージェントは環境を知覚して行動し、環境の状態を変化させる。ここではこれを環境の変換と呼ぶ。タスクを達成するとは、エージェントが環境の変換を繰り返し、環境を何らかのゴール状態へと導くことを意味する。ゴー

†mune@nsc.es.hokudai.ac.jp
‡oyama@nsc.es.hokudai.ac.jp

ル状態までの、環境の変換の連鎖が長いほど、概して難しいタスクであるといえる。

多くの環境の変換の段階全てを、１個のエージェントが行うことが可能であれば、役割分担を行うことの意義は小さいかもしれない。しかし、１個のエージェントが学習できることは有限である。役割分担とは、多くの段階を経てはじめて達成されるようなタスクにおいて、各エージェントがそれら段階のうちの、１つあるいは少数のみを担当し、エージェント集団としては、全ての段階をカバーすることであると考えられる。

工学的な観点からみて、マルチエージェントシステムにおいて、タスク達成という目的に適うような役割分担を実現するには、エージェント及びエージェント間の相互作用をどのように設計すればよいのであろうか。

システムの柔軟性からみると、各エージェントの役割は設計者が予め決めるよりも、エージェントが学習を行うことによって自然に決定されることが望ましい。この場合、各エージェントの正しい振る舞いは未知であるため、学習の枠組みとしては強化学習 [1] のような学習が適していると考えられる。しかしながら、各エージェントにそれぞれの振る舞いに応じて、どれだけの報酬を分配するのかを設計者が予め決めるのは難しい。これは信頼度割当て問題と呼ばれている [2]。

このようなマルチエージェント強化学習における報酬分配の問題に関して、報酬を得たエージェントが他のエージェントに報酬を分配する方法が提案されている [3,4]。この方法においては、エージェントは報酬分配の学習を行動の学習と並行して行っていく。これによって複数エージェントにおけるジレンマ状況が解決できることが示されている。本稿ではこの先行研究をヒントにして、各エージェントに、自身の役割の学習と、エージェント間の報酬のやりとりの学習を同時に行わせることによって、エージェント間の役割分担を自然発生させる可能性について探る。

1.2　報酬分配手法の検討

本小節では、役割分担実現のための報酬分配の枠組みについて検討し、「逆行的な報酬分配手法」について述べる。

先行研究における報酬分配手法の特徴として、最終的にタスクを達成し、システム外部から直接報酬を獲得したエージェント（仮にＺとおく）が、他のエージェント全てに報酬を分配するということが挙げられる。Ｚは報酬分配に関するパラメータを持ち、これはエージェントを一つひとつ区別し、その数だけ用意された実数であり、それぞれにどれだけの割合で報酬を与えるのかを表している。その一方で、Ｚの各エージェントへの報酬分配量は、各エージェントの振る舞いの直接の関数にはなっていない。

役割分担形成という目的のためには、個々のエージェントを区別する代わりに、果たした役割に応じて報酬を分配する方法がありうる。異なるエージェントであっても、同じ役割を果たしたエージェントであれば、同じ量の報酬を与えることは理にかなっている。この場合、タスク達成の連鎖のある段階の役割を果たしたエージェントに対しては、どのエージェントが報酬を分配するべきであろうか。タスクを最終的に達成したエージェントが、他の全てのエージェントの役割・振る舞いを知ることは難しい。そこで、ある段階の役割を担うエージェントに対しては、次の段階のエージェントが報酬を支払うという方法を考えた。本稿では、このような報酬分配手法を、「逆行的な報酬分配手法」と呼ぶことにする。これは、タスク達成への環境の変換の連鎖に対して、報酬分配の流れの方向が逆向きであることを意味する。

逆行的な報酬分配手法の特徴として、報酬分配に関して各エージェントが保持するパラメータの数が少ないことが挙げられる。先行研究手法では、各エージェントが他の全てのエージェントの数だけ、報酬分配に関するパラメータを保持する。一方、逆行的な報酬分配手法の場合、各エージェントは自分のひとつ前の段階（役割）のエージェントだけを考慮する。しかも同じ役割のエージェントは同じように扱う。そのため各エージェントが報酬分配に関して保持し、学習するべきパラメータの数は、最も単純な場合を想定すれば、１個となる。隣り合う段階のエージェント間に、複雑な交渉のルールを考えた場合、より多くのパラメータが必要になる（本研究では各エージェントが最大２個のパラメータを保持するものとした）が、おそらく多くの場合、各エージェントが保持するパラメータの数は、エージェントの総数よりも少なくなるだろう。したがって、逆行的な報酬分配手法では、探索するべきパラメータの集合の大きさが小さくなり、適切な報酬分配学習が速くなるというメリットを期待できる。

1.3　問題設定

逆行的な報酬分配手法が、マルチエージェントシステムにおける役割分担の発生のために実際に有効にはたらくのか否かは不明である。とりわけ、ある段階の役割を担うエージェント（報酬の受け手）と次の段階の役割を担うエージェント（報酬の与え手）の間にどのような相互作用を定義するべきなのかは非自明である。本稿の目的は、報酬の受け手と与え手の間に、「交渉」に対応するような相互作用を導入するべきなのか否かを解明することである（先行研究の報酬分配手法との比較は、本稿の範囲外である）。そのために、逆行的な報酬分配によってエージェン

トが役割を獲得していく様子を表現したシミュレーションモデルを構成し、解析を行う。

2 モデル

本章では逆行的な報酬分配によってエージェントが役割を獲得していくことを確認するためのモデルを構成する。

2.1 概要

はじめに、「環境の状態」を、本モデルでは「財」という言葉で表現する。前述のパンづくりの例で考えると、小麦や粉やパンは、物理化学的な観点からは非常に複雑な「環境の状態」であるが、経済の枠組みでは3種類の「財」として表現される。このような抽象的な概念を用いることで、エージェント間の相互作用の本質的な側面を検討できると考える。

次に、本研究で構成するモデルには空間の概念は無い。モデルの世界には多くのエージェントが存在する。財の種類には番号が付けられており、ある財は別の財に変換可能である。特定の財が“ゴール財”と定義されており、これを直接生成したエージェントに対してシステム外部から報酬が与えられる。

各エージェントは、一定のコスト（負の報酬）を支払い、ある1種類の財を別の1種類の財に変換する。どの財をどの財に変換するのかがエージェントの役割を表しており、エージェントは役割を変えることができる。財の変換の連鎖がつながって、ゴール財生成が行われるために、エージェントどうしはランダムに出会う。このとき、連鎖の前段階のエージェントから後段階のエージェントに財の「引渡し」が行われ、後段階のエージェントから前段階のエージェントに対して報酬が支払われる。

各エージェントは役割以外に、報酬支払い量に関するパラメータを持ち、これも変化させることができる。各エージェントは、多くの報酬を獲得することを目標に、役割と報酬支払い量に関するパラメータの探索を行う[*1]。

さらに、連鎖の前段階のエージェントと後段階のエージェントの相互作用に関して、「交渉」の有るルールと無いルールを定義する[*2]。エージェントが役割分担によって全体として多くのゴール財を生成するために、「交渉」が有益であるのか否かの検討を行う。

[*1] ここでは、価値関数に基づく学習は行わず、よりシンプルな探索を用いた。

[*2] これとは別に対照実験として、報酬支払いが全く行われないというルールも定義する。

2.2 モデルの構成要素

N 個のエージェント： $A_0, A_1, \ldots, A_{N-1}$ と G 種の財：財1、財2、…、財 G が存在する世界を考える。エージェント達は財1から出発して財を変換していき、財 G が作られると、システム外部から報酬が与えられる。

各エージェントは、ある一種の財を、別の一種の財に変換することが可能である。どの財をどの財に変換することが可能であるのかは、予め決められている。財 i を財 j に変換することが可能であるのか否かを、$trans_{i,j} \in \{0,1\}(1 \leq i,j \leq G, i \neq j)$ で表す。財 i を財 j に変換可能である場合 $trans_{i,j} = 1$ とし、変換不可能である場合、$trans_{i,j} = 0$ とする。財 i を財 j に変換するには、コストが $cost_{i,j} \in \mathbb{R}$ だけかかる。

エージェントの能力は、（変換前の財，変換後の財）で表現する。これを、$(material, product)$ と表す。ただし $trans_{material,product} = 1$ でなければならない。財を変換する能力をもたないエージェントも存在し、“無活動”であると呼ぶ。

各エージェントは、他のエージェントから財を譲り受けることがある。このときの相互作用のために、各エージェントは $pay, sell \in [0, PS_{max}]$ の2つの実数値パラメータを持つ。相互作用のルールは後述する。

個々のエージェントのもつパラメータは $(material, product, pay, sell)$ である。

各エージェントは、財 $material$ および財 $product$ 以外の財を持つことは無く、財 $material$ を手に入れた場合それを必ず即時に財 $product$ に変換する。したがってエージェントが持ちうる財は財 $product$ のみである。エージェントは変数 $have \in \{0,1\}$ を持ち、$have = 0$ の場合、エージェントは財 $product$ を持たず、$have = 1$ の場合、エージェントは財 $product$ を持っているとする。

2.3 モデルの動作

初期設定として、N 個の“無活動”のエージェントが生成される。それらの $pay, sell$ はそれぞれ $[0, PS_{max}]$ におけるランダムな値（一様分布）に設定される。

1回の試行は $Period_{max}$ 回の $period$ から成り、各 $period$ は $Time_{max}$ 回の $time$ から成る。各 $period$ の開始時に、エージェントは $have = 0$ に設定される。また、各エージェントは $period$ における累積報酬量 $benefit$ という変数を持ち、これが $period$ 開始時に 0.0 に設定される。各 $period$ では、エージェントはパラメータを変化させない。

各 $time$ では、無活動では無く $have = 0$ であるエージェントが、$A_0, A_1, \ldots, A_{N-1}$ の順に「行動」を行う。行動の内容は次の通りである。なお、以下では ← は計算機プログラムにおける「代入文」を表す。

- $material = 1$ の場合：
  - （財 $product$ の生成にコストがかかる）$benefit \leftarrow benefit - cost_{1,product}$
  - （財 $product$ を持つようになる）$have \leftarrow 1$
- $material \neq 1$ の場合：財 $material$ を持っているエージェントと出会って相互作用し、財を譲り受けることを試みる。

財を譲り受けることを試みる場合、まず相互作用相手の候補を全エージェントから等確率でランダムに選択する。相手になりうるのは、「自分の $material$ と相手の $product$ が等しく、相手が $have = 1$ である」ようなエージェントである。相手の候補として選んだエージェントが、これらの条件を全て満たした場合、次の「相互作用」段階へと進む。満たしていない条件があった場合、何もせずに行動が終了する。

「相互作用」段階では、３種類のルールを考える。しかし、その説明の前に、相互作用を構成する、より小さな「部品」を挙げる。以下では、自分の変数に (me), 相手の変数に、(op) を付して表す。

「支払い」：
- $benefit(me) \leftarrow benefit(me) - pay(me)$
- $benefit(op) \leftarrow benefit(op) + pay(me)$

「引渡し」：
- $have(op) \leftarrow 0$
- $benefit(me) \leftarrow benefit(me) - cost_{material(me),product(me)}$
- $have(me) \leftarrow 1$

各試行では、以下の相互作用ルールのうち、どれか１つが用いられる。
- 「強制」ルール：「引渡し」のみが行われる。
- 「交渉無し」ルール：「支払い」と「引渡し」が行われる。
- 「交渉有り」ルール：$pay(me) \geq sell(op)$ が成り立つ場合に限り（これが「交渉」に対応する）、「支払い」と「引渡し」が行われる。

本モデルにおける「強制」ルールでは、エージェントのパラメータ pay と $sell$ はどちらも用いられない。「交渉無し」ルールでは、pay は用いられるが、$sell$ は用いられない。「交渉有り」ルールでは、$pay, sell$ のいずれも用いられる。

また、$product = G$ であるようなエージェントが行動の結果、$have = 1$ になった場合、このエージェントはシステム外部から報酬を獲得する。
- （財 G を失う）$have \leftarrow 0$
- （報酬を獲得する）$benefit \leftarrow benefit + R$

$period$ 終了時の、各エージェントの $benefit$ は、その $period$ における報酬獲得量の合計、即ち成功度を表している。$period$ が終了すると、エージェントは $benefit$ を最大化するために、パラメータを変化させる。

まず、エージェントのパラメータ $(material, product, pay, sell)$ をまとめて、$params$ と表記する。また、エージェントの今までの経験から比較的優れていると思われるパラメータの記録値を $params_{foot}$, 新しいパラメータに移動するかどうかを決めるために $benefit$ の比較対象とする値を $benefit_{foot}$ と表記する。試行が始まった直後は $params_{foot} = params, benefit_{foot} = -\infty$ に設定する。

ある $period$ が終了した後、各エージェントは次のようにパラメータの探索を行う。

$Case1$：もし $benefit \geq benefit_{foot}$ であれば：
- $params_{foot} \leftarrow params$
- $benefit_{foot} \leftarrow benefit$
- 確率 P_c で $params$ を変化させる。この場合、
  - 確率 L_c で“無活動”なエージェントに、確率 $1 - L_c$ でランダムに新しい役割 $(material, product)(trans_{material,product} = 1)$ になる
  - $pay, sell$ をそれぞれ $[0, PS_{max}]$ の範囲内におけるランダムな新しい値（一様分布）に決める（ただし、$pay, sell$ を変化させない場合の試行も行った）。

$Case2$：もし $benefit < benefit_{foot}$ であれば：
- $params \leftarrow params_{foot}$
- $benefit_{foot} \leftarrow -\infty$

以上のような探索アルゴリズムによって、エージェントは時折新しいパラメータに移動し、そこが今までと同等以上の $benefit$ を獲得できるパラメータである場合はそこに留まり、そうでない場合は前のパラメータに引き返す。

3　シミュレーション

ゴール財生成のために、役割分担が必要になる最も単純なケースとして、次の場合についてシミュレーションを行う。

$G = 3$

$trans_{1,2} = trans_{2,3} = 1$, それ以外の $trans_{i,j} = 0$

この場合のモデルの様相をスケッチすると、図１のようである。各エージェントは、“無活動”という「役割」、または財１を財２に変換する役割（これを以下では“役割１→２”と呼ぶ）、あるいは財２を財 $G = 3$ に変換する役割（同、“役割２→３”と呼ぶ）のいずれかを担う。エージェントはより多くの $benefit$ を得ることを目指して、役割を

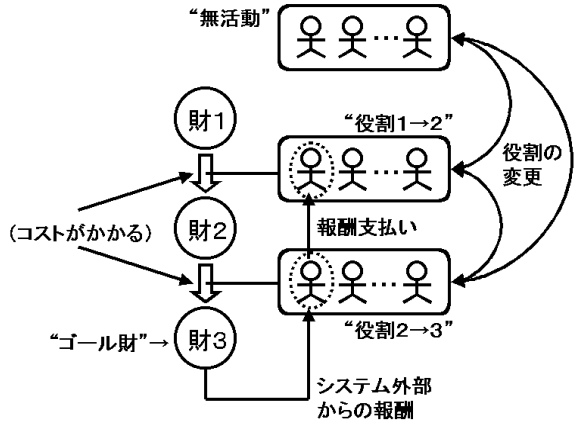

図 1: モデルの様相のスケッチ。“無活動”のエージェント、“役割１→２”のエージェント、および“役割２→３”のエージェントが存在する。エージェントは役割を変更できる。“役割２→３”のエージェントはシステム外部から報酬を受け取り、“役割１→２”のエージェントは“役割２→３”のエージェントから報酬を受け取る。

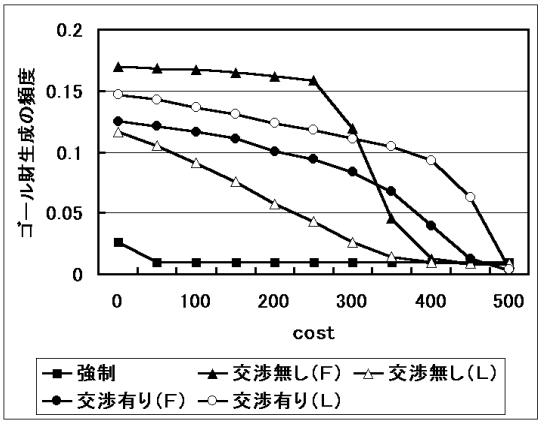

図 2: 各相互作用ルールおよび *cost* 値に対する、ゴール財生成の頻度（1 エージェント・1*time* 当たりの平均ゴール財生成数)。凡例において、（F）は *pay*, *sell* を固定した場合、（L）は *pay*, *sell* の探索を行った場合を表す。

時折変更する。“役割１→２”のエージェントから“役割２→３”のエージェントに、財 2 が引き渡されるときに、後者から前者へ報酬が支払われる。“役割２→３”のエージェントは、引き渡された財 2 を財 $G = 3$ に変換し、システム外部から報酬を受け取る。多くのゴール財が生成されるためには、“役割１→２”のエージェントと、“役割２→３”のエージェントの、エージェント全体に占める比率が適切である必要がある。

ここでは、以下のパラメータは固定して実験を行う。

$N = 200$

$R = 1000.0$

$Period_{max} = 1000$

$Time_{max} = 100$

$P_c = 0.1$

$L_c = 0.5$

$PS_{max} = 1000.0$

各相互作用ルールの有用性を比較するため、それぞれのルールを用いた場合に、ゴール財がどの程度生成されるのかを調べる。「交渉無し」ルールと「交渉有り」ルールについては、各エージェントの *pay*, *sell* を初期条件のまま固定する場合と、これらのパラメータの探索が行われる場合を試みる。

また、ここでは相互作用ルールの有用性に影響を及ぼしうるパラメータとして、財変換のために必要なコストの量にも注目する。コストの量が変わると、“役割１→２”のエージェントと、“役割２→３”のエージェントの *benefit* が、ともに正になることができるような、後者から前者への *pay* の範囲の広さが変わる。そのため、適切な相互作用ルールもまた変わってくる可能性を考慮する。$cost_{1,2} = cost_{2,3} = cost$ とおき、異なる *cost* を試みた[*3]。

全 *period*(= $Period_{max}$ = 1000) について平均した、1 エージェント・1*time* あたりの、平均のゴール財生成数を図 2 に示す。各データ点は、20 回の試行を平均している。

これを見るに、*cost* が比較的低い場合には、「強制」ルール以外のルールでは、1 エージェント・1*time* 当たり、0.05 ~ 0.17 個程度のゴール財が生成されることがわかる。*cost* が高くなるにつれて、どのルールにおいても、ゴール財の生成が少なくなっていく。また、「強制」ルールの場合、全ての *cost* において、ゴール財生成の頻度は低いことがわかる。「強制」ルールが、*cost* に関わらず性能が低かったのは、このルールでは“役割１→２”のエージェントが、コストを支払うばかりで正の報酬を受け取る機会が無いため、この役割を担うエージェントの数が少なくなり、財 1 を財 $G = 3$ にまで変換していく経路が「細く」なったためであろうと推察される。

$cost \leq 250$ においては、「交渉無し」ルールで *pay* を固

[*3] $cost = 500$ の場合、$cost_{1,2} + cost_{2,3} = 1000 = R$ となるため、エージェント達がゴール財を生成しても、そのために必要なコストを差し引くと、正味の報酬獲得量はゼロとなる。

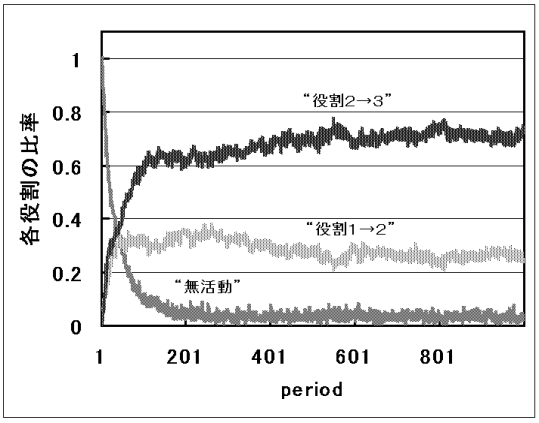

図 3: *period* に対する、各役割のエージェントの比率の変化。

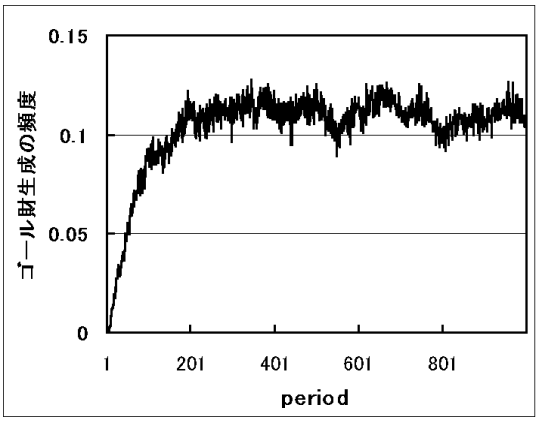

図 4: *period* に対する、ゴール財生成の頻度（1 エージェント・1*time* 当たりの平均ゴール財生成数）の変化。

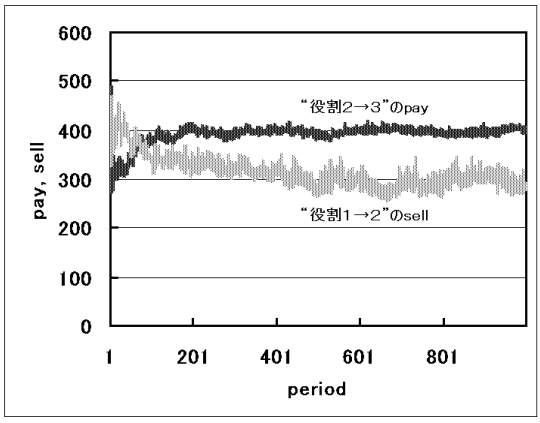

図 5: *period* に対する、各役割のエージェントの *pay*, *sell* の平均値（相互作用に直接関係するもののみ表示）。

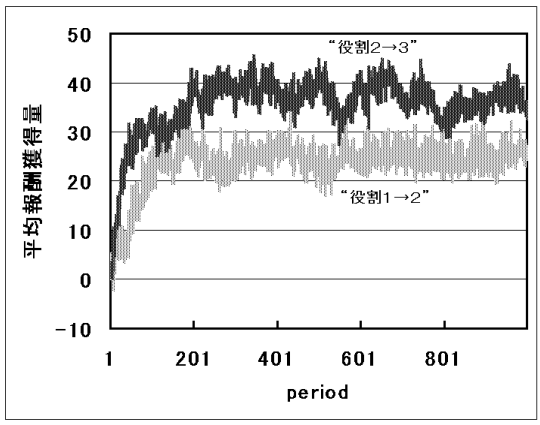

図 6: *period* に対する、各役割のエージェントの平均報酬獲得量（1*time* 当たり）の変化。“無活動”エージェントの平均報酬獲得量は常に 0 なので表示していない。

定した場合に、他の場合よりも多くのゴール財が生成されることが分かる。一方、$350 \leq cost \leq 450$ では、「交渉有り」ルールで *pay*, *sell* に対する探索を行った場合のゴール財生成数が、他の場合よりも多いことが読み取れる。

一例として、$cost = 350$ のもとで、「交渉有り」ルールで *pay*, *sell* の探索を行った場合の、1000*period* における各役割の比率の変化（図 3）、ゴール財生成の頻度（図 4）、各役割のエージェントの *pay*, *sell* 値（相互作用に直接影響するもののみ・図 5）、各役割のエージェントの平均獲得報酬の *period* に対する変化（図 6）を示す。

これらを見るに、最初の約 200*period* で、初期設定で全てを占めていた“無活動”エージェントの比率が減り、“役割 1 → 2”および“役割 2 → 3”のエージェントの比率が増える（図 3）とともに、ゴール財生成の頻度が増大していることがわかる（図 4）。約 200*period* 以降の、“役割 2 → 3”のエージェントの平均の *pay* は、400 程度の値を推移している（図 5）。

ここでは $cost = 350$ であるので、“役割 1 → 2”のエージェントの立場からみると、$cost = 350$ を支払って財 2 を生成し、報酬を 400 程度貰って“役割 2 → 3”のエージェントに引き渡すとすれば、差し引き 50 程度の「利益」が得られる。これと矛盾せずに、“役割 1 → 2”のエージェントは、1*time* 当たり、平均して正の報酬を獲得するようになることに成功している（図 6）。ここで再び図 5 を見るに、“役割 2 → 3”のエージェントの平均の *pay* の値（約 400）は、“役割 1 → 2”のエージェントの平均の *sell* の値（約 300）の幾分上方を推移していることがわかる。この事実は、「交渉」が有ることによって、“役割 2 → 3”の

エージェントの *pay* が低くなり過ぎることが防がれている可能性を示唆している。

4　まとめ

マルチエージェントシステムにおいて役割分担を実現するための、逆行的な報酬分配手法について検討した。エージェント間の相互作用ルールに「交渉」を導入するべきか否かについて解析を行った。その結果、

- 財変換のコストが比較的小さい場合には、「交渉」の無いルールを、報酬支払いに関するパラメータを固定しつつ適用するのが望ましく、
- 財変換のコストが比較的大きい場合には、「交渉」の有るルールを、報酬支払いに関するパラメータを探索しつつ適用するのが望ましい

ことがわかった。特に、財変換のコストがどの程度なのかが、未知である場合には、後者を選択するのが堅実であると考えられる。従って、総合的には、相互作用ルールにおける「交渉」の有用性が示唆されたと結論できる。

参考文献

[1] Sutton R.S., Barto A.G.: Reinforcement Learning: An Introduction. The MIT Press. (1998)

[2] 高玉圭樹：マルチエージェント学習ー相互作用の謎に迫る．コロナ社．(2003)

[3] Shibata K., Ito K.: Autonomous Learning of Reward Distribution for Each Agent in Multi-Agent Reinforcement Learning. Intelligent Autonomous Systems, Vol.6, pp.495-502 (2000)

[4] 柴田克成，真崎勉：多人数ゲームにおける報酬分配学習．計測自動制御学会　システム・情報部門学術講演会 2002 講演論文集, pp.15-20 (2002)