



MACC2001

第 10 回マルチエージェントと協調計算ワークショップ
論文集

武田英明編



ソフトウェア科学会研究会資料シリーズ
MACC2001
第 10 回マルチエージェントと協調計算ワークショップ論文集

2002 年 2 月 1 日発行

編集 武田英明

〒101-8430 東京都千代田区一ツ橋 2-1-2

国立情報学研究所

<http://www-kasm.nii.ac.jp/~takeda/>

電話 03-4212-2543

FAX 03-3556-1916

発行 日本ソフトウェア科学会

〒113-0032 東京都文京区弥生 2-4-16 学会センタービル内

<http://www.jsst.or.jp/jsst/>

電話 03-5802-2060

FAX 03-5802-3007

ISSN 1341-870X No.19

目次¹

はじめに

武田 英明	v
アドバイザリーコミティとプログラム委員	vi

パネルセッション

1. 社会的エージェントの可能性

石田 亨, 中西 英之, 長尾 確, 小野 哲雄	1
--------------------------------	---

エージェントモデル

2. 義務と権利に基づく社会行動の即応制御モデル

大澤 一郎	3
-------------	---

3. BDI Logic の sequent calculus による演繹体系

新出 尚之, 高田 司郎	10
--------------------	----

4. マルチエージェントによるチーム航行支援 (S)

安藤 英幸, 広野 康平, 大和 裕幸	21
---------------------------	----

5. Wizard of Oz 法を使ったマルチキャラクターエージェントインタフェースの評価 (S)

辻本 秀樹, 北村 泰彦, 辰巳 昭治	26
---------------------------	----

エージェントプロトコル

6. エージェント基盤のための協調プロトコル: Agent Platform Protocol

高橋 健一, 鍾 国強, 雨宮 聡史, 峯 恒憲, 雨宮 真人	30
---------------------------------------	----

7. マルチエージェントシステムへの投機的計算導入手法

山本 京司, 佐藤 健	38
-------------------	----

8. モバイルエージェント系における分散制約充足

沼澤 政信, 栗原 正仁, 能登 正人	46
---------------------------	----

9. ユーザ選考に基づく分散コンテンツキャッシュポリシ (S)

中塚 康介, 八槇 博史	54
--------------------	----

コミュニティのためのエージェント

10. ロコミによる分散型情報収集システム

吉田 匡志, 伊藤 雄介, 沼尾 正行	61
---------------------------	----

11. 知識の階層構造を用いた協調的情報推薦 (S)

濱崎 雅弘, 武田 英明, 河野 恭之, 木戸出 正継	69
-----------------------------------	----

¹タイトルに (S) とあるものはショートペーパー、その他はロングペーパー

ロボット

12. RoboCupRescue における最少通話による協調の実現
森本 武資, 河野 健二, 竹内 郁雄 71
13. 異種端末間の協調による高齢者・障害者の移動支援、Robotic Communication Terminals(S)
矢入 (江口) 郁子, 猪木 誠二 78

学習と知識獲得

14. 未来状態の予測を利用したマルチエージェント強化学習法
長行 康男, 柴田 直樹, 伊藤 実 85
15. 環境状況の変化に応じて自己の報酬を操作する学習エージェントの構築
森山 甲一, 沼尾 正行 91
16. 自律学習における CITTA による部分知識の結合 (S)
山川 宏, 宮本 裕二, 馬場 孝之, 岡田 浩之 99
17. ネット情報を使った意思決定のための知識獲得エージェント (S)
藤本 和則, 山本 裕 103

特別セッション [ゲーム理論／意思決定／経済学的アプローチに基づくマルチエージェント]

18. 情報財の開発インセンティブについて
藤山 英樹 107
19. 人工市場における複雑さの発展の分析
和泉 潔 116
20. X-Economy - マルチエージェント経済におけるシミュレーションプラットフォーム
川村 秀憲, 車谷 浩一, 大内 東 122
21. P2P 電子マーケットにおけるエージェント間の提携に基づく売買交渉機構について
服部 宏充, 伊藤 孝行, 新谷 虎松 128
22. U-MART: 経済学と工学をエージェントが結ぶ
喜多 一, 出口 弘, 寺野 隆雄 138
23. エージェントシミュレーションによる「チープ・トーク」モデルの拡張
小山 友介 144
24. エージェント系としてのレプリケータダイナミクスとその制御
國上 真章, 寺野 隆雄 152
25. エージェントシミュレーションによる Prospect 理論と GARCH モデルの関連性の分析
高橋 大志, 寺野 隆雄 158
26. 複数財オークションについて
松井 知己, 渡辺 隆裕 166
27. セキュアマルチエージェント動的計画法の設計と組合せオークションへの適用
鈴木 幸太郎, 横尾 真 171

はじめに

プログラム委員長 武田 英明

本論文集は日本ソフトウェア科学会「マルチエージェントと協調計算研究会」の第 10 回ワークショップ(第 10 回マルチ・エージェントと協調計算ワークショップ, MACC2001)で発表されたものを集めたものである。MACC2001 においては投稿された論文はプログラム委員による査読を受けて、著者による加筆・修正ののち発表、本論文集に掲載というプロセスを経ている。本年度は、通常セッションにロングペーパー 9 件、ショートペーパー 7 件、特別セッション 10 件の発表が行われた。

MACC2001 は金沢のラポート兼六を中心に 2001 年 11 月 16 日, 17 日に開催された。参加人数は約 60 人で、基本的に全員泊りがけで行った。

初日は前日より金沢大学で開催していた電子情報通信学会人工知能と知識処理研究会と共同企画でパネルを行った。その日は懇親会を兼ねた夕食の後、発表者以外の参加者の 30 人弱のポジションペーパー発表が行われた。ポジションペーパー発表とはいえ、興味深い発表もあり、活発な質疑が行われた発表もあった。

翌日からは 2トラックの平行セッションとして行った。今回は、特別セッションに多数の論文が集まったため、基本的に 1つのトラックを特別セッションとして行った。

通常セッションは、ロングペーパー、ショートペーパーを織り交ぜ、「コミュニティのためのエージェント」、「ロボット」、「エージェントモデル」、「エージェントプロトコル」、「学習と知識獲得」というセッションに分けて行った。

特別セッションは「ゲーム理論／意思決定／経済学的アプローチに基づくマルチエージェント」というテーマで 10 件

の発表が行われた。時間帯によっては通常セッションよりむしろ特別セッションの方に人があつまり、立ち見ができる状況であった。従来の MACC の発表者が人工知能を中心とする情報工学の分野であったの対して、この特別セッションの発表者はセキュリティの研究者や経済学の研究者など幅広いものであり、この分野が広範な範囲の人々の興味をひきつけており、今後の発展の可能性を示していた。

このようにマルチエージェントの分野は単に人工知能あるいは情報工学の特定の興味でなく、広範な興味・話題を含みようになっている。このような認識のもとで MACC は来年度は電子情報通信学会人工知能と知識処理研究会と共同で、エージェント／マルチエージェントの話題を広く含むような会議として開催されることになった。同じく 11 月中旬に函館での開催が予定されている。

本ワークショップ開催にあたってはプログラム委員のみなさまの協力のもとで成功裏に開催することができた。ここに感謝の意を表する。

Preface

Hideaki Takeda, 国立情報学研究所, National Institute of Informatics.

アドバイザーコミッティ

石田 亨 (京都大学)
奥乃 博 (京都大学)
所 真理雄 (ソニー)
中島 秀之 (産業技術総合研究所)
西田 豊明 (東京大学)
本位田 真一 (国立情報学研究所)

プログラム委員

[委員長] 武田 英明 (国立情報学研究所)
伊藤孝行 (北陸先端科学技術大学院大学)
和泉 潔 (産業技術総合研究所)
磯崎 秀樹 (NTT)
大澤 一郎 (産業技術総合研究所)
大沢 英一 (はこだて未来大学)
大澤 幸生 (筑波大学)
小野 典彦 (徳島大学)
北村 泰彦 (大阪市立大学)
佐藤 理史 (京都大学)
榎木 哲夫 (京都大学)
島津 秀雄 (NEC)
菅沼 拓夫 (東北大学)
菅原 俊治 (NTT)
角 康之 (ATR)
高間 康史 (東京工業大学)
垂水 浩幸 (香川大学)
津田 和彦 (筑波大学)
中内 靖 (防衛大学校)
長尾 確 (名古屋大学)
西村 俊和 (立命館大)
橋田 浩一 (産業技術総合研究所)
平山勝敏 (神戸商船大学)
藤本 和則 (NTT)
松原 繁夫 (NTT)
本村 陽一 (産業技術総合研究所)
八槨 博史 (京都大学)
横尾 真 (NTT)
吉田 哲也 (大阪大学)

パネル：社会的エージェントの可能性

石田 亨† 中西 英之† 長尾 確‡ 小野 哲雄††

エージェント研究の流れは、コンピュータインタフェースの一形態として人間のコンピュータ利用を支援する「個人的エージェント」から、社会的役割を持ち、人と人のインタラクションに介在する「社会的エージェント」へと向かっているように思われる。本パネルでは、ソフトウェアエージェントからロボットに至るまでの、我が国で最新の研究を紹介し、「社会的エージェント」という新しい研究分野に定義を与え、その可能性を議論する。

1 パネルプログラム

- 司会 石田 亨
- 社会心理学実験報告：エージェントは人間関係を操作できるか 中西英之
- 社会的マシンとしての会話ロボット 長尾確
- Robovie：コミュニケーションにおける身体性と関係性 小野哲雄

2 各講演概要

2.1 社会心理学実験報告：エージェントは人間関係を操作できるか

中西英之 (京都大学社会情報学専攻)

ソフトウェアエージェントが人間関係に及ぼす影響を明らかにする第一歩として、エージェントが人間関係を操作でき

るかどうか確かめる実験を行なった。この実験はバランス理論にもとづいて行われた。バランス理論によると、2 人の人間がある対象に同じ心情(好意もしくは敵意)を持つ場合の方が、片方は好意を持ち、もう片方は敵意を持つ場合よりも、2 人の間の関係が良くなる。エージェントと人間 2 人の関係にこれを当てはめた。

実験ではまず、エージェントが同意の態度を示すことで人間に好意を持たれ、逆に反意の態度を示すことで敵意を持たれることを確かめた。次に、バランス理論が人間 2 人とエージェントの関係に成り立つかどうかを観察した。バランス理論が成立するとすれば、エージェントが両方の人間に同意や反意の態度を取る場合の方が、片方にだけ同意して、もう片方に反意する場合より、人間同士の関係は良くなる。

実験の結果、人間同士の会話が無いときは、人間関係をエージェントが操作できた。しかし、少しでも会話があると、エージェントは人間関係を操作できなかった。少しの会話がエージェントの能力を失わせた原因を探るため会話分析を行ったところ、会話を通して人間 2 人が互いに共感することが原因であると判明した。

2.2 社会的マシンとしての会話ロボット

長尾確 (日本IBM東京基礎研究所)

IBM 東京基礎研究所では、新たなヒューマンインタフェースとして会話ロボットに関する研究を進めている。IBM の取り組むロボットは、手足の動きなど外観の機能や動きを追求するのではない。「人間にとって役に立つ、便利な助手のような存在」として利用できるようにするための仕組み、つまり知能の部分にフォーカスを当てている。人間とのコミュニケーションの手段は非常に身近な「会話」である。しかしながら、人間が機械に対して話しかける行為に抵抗を感じるという問

Panel: Potential of Social Agents

† Toru ISHIDA, Hideyuki NAKANISHI 京都大学社会情報学専攻 Department of Social Informatics, Kyoto University

‡ Katashi NAGAO 日本アイ・ビー・エム株式会社 東京基礎研究所 IBM Research, Tokyo Research Laboratory

†† Tetsuo ONO 公立はこだて未来大学, Future University Hakodate

題をまずはクリアしなければ役には立たない。人間は無意識のレベルで、存在感があり、動作する対象が自分に注意を向けていると感じ取ると、それは「自分とコミュニケーションする対象なのだ」と直感的に認識していると思われる。IBM は直感的で臨場感のある「会話」によって、人間と機械がスムーズなコミュニケーションを行なうために、「ロボット」という入れ物を利用している。ロボットは、話しかけられると、ユーザーの意図を理解し、必要な情報を集めて自動的に処理し、適切な応答の生成を試みる。さらにはユーザーと行ってきた会話の履歴を記憶する能力を持たせることにより、ユーザーの嗜好にあった行動をロボットが自発的に行なってくれるようになるだろう。また、ユーザーに特化し同調するような仕組みとして感情を理解できるシステムもあらかじめ組み込んでおけば、ユーザーがうれしいとき、ロボットもうれしいし、楽しいとロボットも喜んでくれるだろう。ロボットが知性と感情を持ちあわせるようになったとき、便利さだけでなく、人間を癒してくれる存在となるだろう。人間との会話や知識を処理するインテリジェンスを持たせたとき、ロボットは人間にとって本当に信頼のおけるパートナーになり得るだろう。また、ロボットは、コミュニティを媒介し、人同士のコミュニケーションを促進するシステムとしても機能する。たとえば、会話ロボットは遠隔地にいるもの同士が非同期でコミュニケーションする場合に、物理的な存在感を備えた、擬似的な対面性を実現することができる。また、長い経験を通じて個人情報を獲得したロボットは、その人の外在化された記憶あるいはパーソナリティを持つシステムとして機能できるようになるだろう。これは、ロボット（あるいはエージェント）が文字通り人間の代理人になるということであるが、もしロボットの信用や責任という問題がうまく技術的あるいは社会的に解決できれば、人間は自分のコピーあるいは分身を作って同時に複数の作業ができるようになるだろう。そのようなロボットを社会的マシンと呼び、その可能性と課題について議論する。

2.3 Robovie: コミュニケーションにおける身体性と関係性

小野哲雄 (公立はこだて未来大学 / ATR メディア情報科学研究所)

近年、エージェントやロボットの社会性に関する議論が行われている。本発表では、コミュニケーションにおける「関係

性」と「身体性」に注目し、社会的ロボットの実現可能性とその研究方法について述べる。

エージェントやロボットなどの人工物が社会的存在となるためには、人と自然で円滑なコミュニケーションを実現しなければならない。しかし、いまだに人は、エージェントやロボットとのインタラクションに違和感を持ってしまう。この理由は、これまでのコミュニケーション研究が言語的側面だけを重視してきたことによる。本発表では、自然なコミュニケーションを実現するには、その成立の基盤をなす「関係性」と「身体性」に注目する必要がある、さらにそれらの機能をシステムに取り込む必要があることを主張する。

本発表では、2 つの研究事例を紹介する。まず、Ono [1] は、人とロボットの間に関係性を構築するためのインタフェースとして、エージェントマイグレーションを用いた手法を提案した。具体的には、ユーザとインタラクトしていた携帯端末上のエージェントがロボットへ移動することにより、両者の間に関係を構築することを目指したものである。実験の結果、エージェントが移動したときユーザはロボットの発話を理解できるが、移動しないときは理解できないことが明らかとなった。

次に、Ono [2] は、コミュニケーションにおける空間表象の伝達と身体動作の関係に注目し、共創対話のモデルを提案した。本モデルは、身体と同調的動作による対話者間の関係の構築、および、この関係に基づく情報の伝達のメカニズムを理解するためのモデルである。ヒューマノイド型ロボットを用いた実験の結果、人は身体と同調的な動作により円滑な情報伝達を行っていることが明らかとなった。

これらの研究成果から、コミュニケーションにおける関係性と身体性の重要性が確認できた。さらに、これらの結果は、社会的ロボットの実現方法に示唆を与えるものである。

- [1] T. Ono, M. Imai (2000). Reading a Robot's Mind: A Model of Utterance Understanding based on the Theory of Mind Mechanism, AAAI-2000, pp. 142--148.
- [2] T. Ono, M. Imai, H. Ishiguro (2001). A Model of Embodied Communications with Gestures between Humans and Robots, CogSci-2001, pp. 732--737.

義務と権利に基づく社会行動の即応制御モデル

大澤 一郎

A new computational model of socially interactive systems driven by obligations and rights is proposed. Obligations and rights are explicitly represented with the operators “O” and “R”, and the internal computation is interactively executed on a reflective architecture based on recall and observation inside. We implement a prototype system to show its performance on some social domains such as business transactions, secure information handling, and driving on the road.

1 はじめに

コンピュータ・システムが社会のあらゆるところに入り込み、政治、経済、金融から、一般の人々の生活までもが、世界中に張り巡らされたコンピュータ・ネットワークによって密接に結び付けられつつある。このような状況下では、ネットワーク内で活躍するエージェント・システムも社会的あるいは組織的に守られた特殊な存在ではなく、社会人と呼ばれる成人のように社会で一人前に行動できなければならない。しかしながら、これまでの大半のエージェント・システムは、BDI モデルで代表されるような、欲求ベースで行動する幼児のような存在でしかなかった。

近年、社会的行為における義務と権利の重要性が明らかになるにつれ、この分野で多くの研究が行われつつある。様相論理の一種である義務論理の多くは、義務と

許可を双対なものとみなして体系化しようとして[4]、義務論理のパラドックスを生じさせていた。Castelfranchi [2]は権利を社会的コミットメントに基づいて派生するものと考えたが、Norman [5]は権利を明示的に表現し、権利の概念を柔軟に扱おうとした。Alonso [1]は、義務と権利の関係に着目し、権利の背後にある周囲の義務を含めて権利を定義しようとしている。このような権利の明示的表現という論理的な研究の流れに対応して、Sloman [3]らのグループは、データベースにおけるセキュリティ管理を一般化して、役割(ロール)に基づく義務と権利をオブジェクト指向モデルに導入している。また、Wagner [8]はエージェント指向モデルに義務と権利の枠組みを導入しようとしている。

本研究では、このような研究動向を踏まえ、義務と権利に基づいて社会的に自律動作するインタラクティブシステムの内部計算モデルを提案し、実際に実装したプロトタイプによる社会的行動の内部シミュレーションを示す。以下では、2 節でシステム内部における情報の表現形式を説明し、3 節で基本となる計算モデルを集合的概念に基づいて提案する。そして、4 節でマルチエージェントによる実装と、5 節でプロトタイプによるシミュレーションを紹介する。

2 義務と権利の表現

2.1 BDI モデルの言語拡張

義務と権利に関する情報を明示的に扱うために、人工知能における BDI モデル[7]を言語拡張する。義務を $O(\text{obligation})$ で表現し、権利を $R(\text{right})$ で表現する。

A reflective computational model of socially interactive systems

Ichiro Osawa, AIST 情報処理研究部門

基本的な BDI モデルでは、システムの内部状態を信念 B (belief) と欲求 D (desire) と意図 I (intention) によって表現する。信念 B がシステムの信じている情報で、欲求 D がシステムの望んでいる状態あるいは事象、意図 I がシステムの決意した状態あるいは事象である。例えば、システム s が x という命題を信じていて、 y という状態を望み、 z という事象を意図している場合を以下のように表現する。

Example 1

$$Bs.x \wedge Ds.y \wedge Is.z$$

欲求 D は実現すれば望ましい状態であり、将来において成り立つ可能性があると思っていれば、他の欲求と相反していても良い。一方、意図 I は実現を確信して身を委ねるような状態あるいは事象であり、将来において必ず成立すると信じていなければならない。

義務 O は、あるシステム s がシステム t に対して状態あるいは事象の実現の責務を負っていることを表現する。この義務の表現を用いれば、店主が客から代金を受け取った場合に商品を渡さなければならないという規則を次のように記述することができる。但し、[行為] は $done$ (行為者, 行為) の略記である。

Example 2

$$Os, t, [give, t, x]$$

$$\leftarrow Ds.[sell, t, x] \wedge Bs.price(x, p) \wedge Bs.[take, t, p]$$

なお、基本的な BDI モデルでは、意図 I は行為の実行を意図する場合と、状態の達成を意図する場合の両方に用いられるが、本モデルでは意図 I を行為の実行意図のみに用い、状態達成の意図は自己に対する義務 $O_i, i.x$ で表現している。

権利 R は、システム s に事象の実行権があることを表現する。例えば、店主は客に商品売る義務があれば、その客から代金を受け取る権利があるという規則を以下のように記述できる。

Example 3

$$Rs.[take, t, p]$$

$$\leftarrow Os, t, [sell, t, x] \wedge Bs.price(x, p) \\ \wedge Bs.\neg[take, t, p]$$

2.2 メッセージの解釈

オブジェクト指向システムでは、システムがメッセージを受信すると、受信メッセージに対応するメソッドが自動的に起動され、メソッドのプログラムに従ってメッセージの処理が実行される。メッセージの基本的な送受信機構に義務や権利の概念は含まれていないので、送信者によって異なるメッセージ処理を行うようにするには、メソッドのプログラム内に送信者情報に基づいた分岐処理を内在させておかなければならない。

それに対してこのシステムでは、明示的に定義してある規則に基づいて、受信したメッセージを義務(あるいは情報)として解釈し、義務の履行という形式でメッセージの処理を実行する。例えば、システム tom からの質問メッセージに対して必ず回答しなければならないシステムの場合、以下の規則を定義しておく($Ms.内容@t$ は t から s へのメッセージを表す)。

Example 4

$$Oi, s, [ans, s, p] \leftarrow Mi.[ans, s, p]@s \wedge s = tom$$

3 自己反映型即応計算モデル

3.1 計算モデル

信念 B 、欲求 D 、意図 I 、および義務 O と権利 R に基づいてシステムをインタラクティブに動作させるために、その計算機構として自己反映型[6]の即応システム RRS を用いる。この節では、この計算機構のベースになっている計算モデルを集合論的な概念に基づいて定義する。

自己反映型即応システム RRS は、時々刻々と変化する状態 RS_t と想起規則 RR とから構成され、時刻 t における状態 RS_t は、観測 SO_t と記憶 SM_t と想起 SR_t の 3 状態から構成される。

Definition 1 (自己反映型即応システム RRS)

$$RRS = (RS_t, RR) \quad [t = 0, 1, 2, \dots]$$

$$RS_t = (SO_t, SM_t, SR_t)$$

時刻 t におけるシステムの観測としては、外部世界からのメッセージ M あるいはイベント情報 E 、若しくはシステム自身がその時点で想起している信念 B と欲求 D と意図 I と義務 O と権利 R に関する想起情報 SR_{t-1} の中から、まだ未観測のものを一つだけ取捨選択することができる。すなわち、システムの観測対象で

ある世界 W_t は、メッセージ M とイベント情報 E から成る外部世界 OW_t と、想起 SR_{t-1} に基づく内部世界から構成されており、時刻 t における観測 SO_t は次のように表現できる。但し、 $:$ はことば同義で、右側の集合に含まれる要素の種類 (タイプ) を拡張 BDI モデルのオントロジーに基づいて補足的に表現している。例えば $S : M + E$ は、集合 S の要素が M あるいは E の情報であることを表している (尚、省略形として $M + E$ の代わりに ME 、 $B + D + I$ の代わりに BDI 、 $O + R$ の代わりに OR を用いている)。

Definition 2 (観測 SO_t)

$$W_t = OW_t \cup SR_{t-1} : ME + BDI + OR$$

$$SO_t \subset W_t - SM_{t-1} : ME + BDI + OR$$

$$|SO_t| \leq 1$$

記憶 SM_t は、直前の記憶 SM_{t-1} をベースに計算する。 SM_t は、 SM_{t-1} の情報をほぼ全て継承するが、メッセージ M の場合は、そのメッセージから想起規則 RR で想起され未観測のものがあるときだけ継承される。解釈の終了したメッセージを記憶から抹消するためである。そして観測 SO_t を加えることで時刻 t における記憶 SM_t が計算できる。

Definition 3 (記憶 SM_t)

$$SM_t = SO_t \cup (SM_{t-1} - M)$$

$$\cup \{m | m \in SM_{t-1} \cap M$$

$$\wedge |RR(m, SM_{t-1}) - SM_{t-1}| > 0\}$$

$$SM_t : ME + BDI + OR$$

想起規則 RR は、記憶 SM_t 内の情報に基づいて想起する情報を求める規則である。規則は大きく分類して4つのタイプがある。(1) メッセージ M から信念 B あるいは義務 O を想起する規則 RR_1 、(2) 欲求 D から別の欲求 D 、あるいは意図 I 、義務 O 、権利 R を想起する RR_2 、(3) 意図 I から実行プロセスを想起する RR_3 、そして(4) 義務 O から欲求 D あるいは権利 R を想起する RR_4 である。 RR_1 でメッセージを解釈し、 RR_2 はいわゆるトップダウンのプランニング、 RR_3 が意図に対応する行為の実行、 RR_4 が義務 O から欲求 D や権利 R を想起する。

Definition 4 (想起規則 RR)

$$RR_1 : M \times SM_t \rightarrow B + O$$

$$RR_2 : D \times SM_t \rightarrow D + I + O + R$$

$$RR_3 : I \times SM_t \rightarrow E$$

$$RR_4 : O \times SM_t \rightarrow D + R$$

$$RR = RR_1 + RR_2 + RR_3 + RR_4$$

$$RR : (M + D + I + O) \times SM_t \rightarrow E + BDI + OR$$

以上のことから時刻 t における想起 SR_t は、記憶 SM_t と想起規則 RR から以下のように定義できる。

Definition 5 (想起 SR_t)

$$SR_t = \bigcup_{x \in SM_t \cap (M \cup D \cup I \cup O)} RR(x, SM_t)$$

$$SR_t : E + BDI + OR$$

ここで強調しておきたいことが一つある。このモデルの本質的な点であるが、システムが想起していても観測していない情報は、システムに認識されておらず、システムの動作に全く影響を与えていないということである。 RR_3 による行為の実行だけは、実行それ自体をシステムが観測していなくても外界に何かしらの影響があるので間接的な情報の観測によってシステムは影響されるかも知れない。しかしながら、それ以外の信念 B 、欲求 D 、意図 I 、義務 O 、権利 R の想起に関しては、直にその想起を観測しない限りはシステムに影響を与えることがない。

3.2 記憶の自動調整

システムの信念 B は時々刻々変化している。ある時、活動用のエネルギーが不足してきたという信念に基づいてエネルギーの補充欲求が生じたとしても、その後の信念の変化によって、活動用のエネルギーが不足しているという信念が消えてしまうことがある。このような場合には、エネルギーの補充欲求も消滅してほしい。この思考の自動調整機能は以下のように簡単に実現できる。

Definition 6 (欲求・意図・権利の活性状態)

$$x \text{ is active} \Leftrightarrow x \in SR_t \cap SM_t \cap (D \cup I \cup R)$$

システムの欲求 D 、意図 I 、権利 R は、想起規則に基づいて想起され、それが観測されることで活性化し、システムの動作に影響を与える。しかしながら、 SM_t 内の信念等の変化によって、活性化していた欲求 D 、意図 I 、権利 R の想起がなくなってしまうことがある。この場合、観測によって SM_t 内に加えられている欲求 D 、意図 I 、権利 R に関して、以下の定義にしたがって活性状態でなくなったものを継承しないように

する。

Definition 7 (修正 SM_t)

$$\begin{aligned} SM_t = & SO_t \\ & \cup SM_{t-1} \cap (E \cup B \cup O) \\ & \cup \{x | x \in SM_{t-1} \cap (D \cup I \cup R) \wedge x \text{は} active\} \\ & \cup \{x | x \in SM_{t-1} \cap M \\ & \quad \wedge |RR(x, SM + t - 1) - SM_{t-1}| > 0\} \end{aligned}$$

4 マルチエージェントアーキテクチャ

4.1 基本エージェント

前節で述べた自己反映型即応システムを効率よく実装するために、表示と観測に基づいて各エージェントが自律動作するマルチエージェントアーキテクチャ [9] を用いる。各エージェントは、自己および他エージェントの表示している情報を常時観測し、エージェント毎に定義されている規則に基づいて表示情報の更新計算を行う。システム内で変化する情報は全体の僅かな割合なので、この計算は非常に素早く実行できる。全てのエージェントの表示情報が固定した時点で、自己反映型即応システムの一時刻における計算が終了する。

Definition 8 (エージェント)

エージェント $A = (\text{観測} A_O, \text{規則} A_R, \text{表示} A_D)$

各エージェントはローカルに情報を保持し、保持している情報が全てそのまま表示される。自己の表示している情報は全て観測できるが、他エージェントの表示情報に関しては予め観測したいものを宣言しておく。すなわち、観測したい情報の種類と情報源を予め観測 A_O として定義しておく。例えば、*memory* エージェント内の意図情報を観測したい場合には以下のように定義しておく。

Example 5

$[observing, Ix.y@memory]$ ※ x と y は変数

この定義により、*memory* エージェントの表示 A_D に $Ia.[eat]$ 等の情報が表示されていれば、その情報を観測することができる。

Definition 9 (規則 A_R)

規則 A_R : 表示情報 $\times$ 非表示情報 $\rightarrow$ 表示情報

規則 A_R は、観測した自己および他エージェントの表示情報から自己の表示 A_D を更新計算する。各規則は 2 つの入力項と 1 つの出力項から構成される。入力

項の 1 つ目は表示されていない情報で、2 つ目は非表示でなければならない情報である (他エージェントの表示情報は「情報@ エージェント」という表現形式によって自己の表示情報と区別している)。出力項には、入力項の条件が満たされた時に自己の表示 A_D に付け加えるべき情報を定義しておく。以下に示した例は、食べたいという情報があり、食べる権利のあるものに関する情報がエージェント m から観測できない場合に、食べる権利のあるものを欲求するという情報を自己の表示 A_D に付け加えるという規則である。

Example 6

$\{Dx.[eat]\} \times \{Rx.z * [eat, z]@m\} \rightarrow Dx.y * [eat, y]$

4.2 エージェントの拡張

規則 A_R を拡張し、自己の表示情報を更新するだけでなく、他エージェントに直接作用して他エージェントの表示情報を更新できるようにしてある。すなわち、出力項の情報定義を「情報@ エージェント」という形式で記述しておくことで、入力項の条件が満たされた時に出力項の情報が別のエージェントの表示情報として用いられる。これはエージェントのモジュール性を損なうようにも思えるが、オブジェクト指向のメッセージに代わるものとして利用することができる。

そして、人間の長期記憶あるいはデータベースのように持続性をもって蓄えられる情報を処理するために、特殊なエージェント m を用意した。この m エージェントに対して、「+情報@ m 」という形式で直に表示情報を送ると、内部に情報が書き込まれ持続的に表示される。その後に情報を抹消したい場合は「-情報@ m 」という形式で m エージェントに抹消希望の情報を送る。

4.3 自己反映型即応システムのアーキテクチャ

自己反映型即応システム RRS は、 m エージェントを含む 12 の主要エージェントから構成される (図 1)。*sensor* エージェントは、外部からのメッセージあるいはイベント情報、若しくはシステムが想起している BDI+OR 情報の中からまだ未観測のものを観測して m エージェントに書き込んでいく。*stm* エージェントは *sensor* エージェントが観測したメッセージを解釈終了まで一時的に保持する。メッセージ解釈は

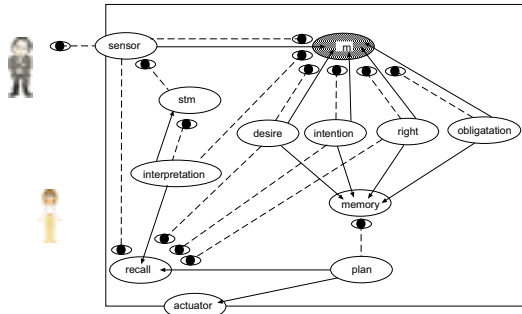


図 1 アーキテクチャ

interpretation エージェントが想起規則 RR_1 に基づいて行い、結果を *recall* エージェント内に表示する。*desire* エージェントと *intention* エージェントと *right* エージェントは欲求 D と意図 I と権利 R を管理し、活性状態のものを *memory* エージェント内に表示し、活性状態でなくなったものを m エージェント内から抹消する。*obligation* エージェントは義務 O を管理し、未達成の義務を *memory* エージェント内に表示し、達成した義務を m エージェントから抹消する。*plan* エージェントは *memory* エージェント内の BDI+OR 情報を観測し、想起規則 (RR_2 と RR_3 と RR_4) に基づき想起された情報を *recall* エージェントあるいは *actuator* エージェントに表示する。

4.4 sensor エージェント

sensor エージェントは、外部からのメッセージとイベント情報、およびシステムが想起している BDI+OR 情報を観測し、その中からまだ m エージェントに書き込まれていないものを各单位時間毎に最大で一つ取捨選択し、 m エージェントに書き込んでいく。この観測制限は、システム内に雑多なシンボル情報が一度に流れ込んで解釈処理等が混乱するのを避けるため、どの情報を選択するかはこのエージェントに託されている。以下は現在の取捨選択の優先順序である (同優先度内ではランダム)。

1. 外部の注目情報源 (話相手等) からの情報
2. 内部で想起された BDI+OR 情報
3. 外部からの情報

5 義務と権利に基づく社会行動システム

5.1 プロトタイプ・システム

これまでに、本システムにおける情報表現 (2 節)、計算モデル (3 節)、実装方法 (4 節) に関して説明してきた。この節では、SUN の SICStus Prolog (v.3.8.5) 上に実装したプロトタイプシステムでの実験に基づき、義務と権利をベースにした社会行動の即応制御がどのように実現できるかを概説する。

例題として、このプロトタイプシステムに対し、Tom という人物から Jo という人物の血液型を尋ねられたという簡単な状況を考える。システムは Tom に対する問いに対しては回答する義務があり、Jo はシステムが管理している病院の患者なので、システムは *doctor* 権限があればデータベース DB に問い合わせて調べることができる。

5.2 想起規則

この例題で必要となる主要な想起規則は、以下の想起規則と次節 (5.3.) で述べる役割に基づく想起規則の二種類になる。

Example 7

想起規則 (RR_1)

$$(1) Oi, tom.[ans, tom, q] \leftarrow Mi.[ans, tom, q]@tom$$

想起規則 (RR_2)

$$(2) Dx.(y : p) \leftarrow Dx.[ans, z, y : p] \wedge \neg Bx.(y : p > a)$$

$$(3) Dx.role(doctor) \leftarrow Dx.(y : p) \wedge Bx.patient(y) \wedge \neg Rx.role(doctor)$$

$$(4) Ix.[ask, DB, q] \leftarrow Dx.[ask, DB, q]$$

(1) の想起規則は、Tom からの問い合わせに対する回答義務を表現している。(2) は、対象 y の属性 p 値を対象 z に報告したい欲求があるが、その値を知らない場合に、その値を知りたいという欲求の想起で、(3) は、対象 y の属性 p 値を知りたいという欲求があり、 y は患者であるが、*doctor* 権限がない場合に *doctor* 権限が欲しくなるという欲求の想起である。最後の (4) は、データベース DB に問い合わせたいという欲求があれば、問い合わせようという意図の想起規則である。

その他に、本論文では範囲外としてあまり言及してこなかったが、義務達成時に義務情報を抹消するための

規則が必要になる。

Example 8

- (5) $\neg O x, y, [ans, y, q]$
 $\leftarrow O x, y, [ans, y, q] \wedge [ans, y, q > a] @ x$

5.3 役割に基づく情報管理 (役割エージェント)

義務と権利に関する多くの規則は、部長や医者等の役割 (権限) という枠組みで管理するのが望ましい[3]。このシステムでは、役割に応じて専門の役割エージェントを用意し、役割に応じた義務と権利の定義が可能になっている。さらに、役割によって得た情報を役割エージェント内で管理することで、異なる役割での越権的な情報処理を防いでいる。以下の想起規則は *doctor* という役割エージェント内で定義されている。

Example 9

想起規則 (RR_2)

- (6) $Rx/doctor.[ask, DB, y : p]$
 $\leftarrow Dx.(y : p) \wedge Bx.patient(y)$
 (7) $Ix/doctor.[ans, z, q > a]$
 $\leftarrow Dx.[ans, z, q] \wedge Bx/doctor.(q > a)$

想起規則 (6) は、対象 y の属性 p 値を知りたい欲求があり、 y が患者であれば、データベース DB に問い合わせ権利の想起である。(7) は、対象 z に質問 q の回答を行いたい欲求があり、*doctor* 権限においてその回答を知っていれば、*doctor* という役割条件で回答を行う意図が想起されるという規則である。

5.4 シミュレーション (情報管理)

図 2 に実行履歴を示す。観測情報の右側は情報源で、想起の○が未観測の情報、●が観測済みの情報を表している。最初 (時刻 320) に Jo が患者であるという情報を観測している。時刻 331 で Tom からの問い合わせメッセージ (Jo の血液型) を受けた。時刻 332 で Tom からのメッセージを回答義務として解釈し、時刻 333 以降で *doctor* 権限欲求を想起している。時刻 416 で *doctor* 権限を得たので DB への問い合わせを実行し、時刻 423 に情報 (A 型) を得ている。その結果、Tom への回答意図が想起され、時刻 424 でその意図を認識し、*doctor* 権限で Tom への回答を行っている。

6 結論と今後の方向

義務と権利に基づいて社会的に自律動作するインタラクティブシステムの内部モデルを提案した。表現形式として BDI モデルを拡張し、計算モデルとして自己反映型即応計算モデルを提案した。

マルチエージェントベースのプロトタイプ実装により、例題として取り上げた *doctor* 権限による質疑応答以外に、商取引での基本的な売買における義務と権利の問題や、自動車の運転における左方優先義務を順守した直進等の例題シミュレーションでシステムがうまく動作することをこれまでに確認している。

Example 10

店主の想起規則 (RR_4)

$Rx/shop.[take, y, m]$

$\leftarrow O x, y, [sell, y, a] \wedge Bx.price(a, m) \wedge \neg [take, y, m] @ x$

運転手の想起規則 (RR_2)

$Rx/driver.[move] \leftarrow Dx.[go] \wedge \neg Bi.exist(c, L)$

今後は、システムに複数の役割がある場合の役割同士間の相互関係モデル、および義務の達成条件の一般的なモデル、そして達成不可能な場合の責任 (義務) のモデルを構築し、システムに実装していきたいと考えている。また、Sloman 流の役割ベースの義務と権利の管理方式と Wagner のダイアグラムを融合させ、義務と権利のダイアグラムから想起規則を自動生成することで、既存の義務と権利に関する外部モデルとこの論文で提案した内部モデルをリンクさせていきたい。

参考文献

- [1] Eduardo Alonso. Right and coordination in multi-agent systems. In *UKMAS'98*, 1998.
- [2] Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *In Proc. ICMAS-95*, pp. 41–48, 1995.
- [3] Emil C. Lupu and Morris Sloman. Towards a role based framework for distributed systems management. *Journal of Network and Systems Management*, Vol. 5, No. 1, pp. 5–30, 1997.
- [4] J. J. Ch. Meyer and R. J. Wieringa, editors. *Deontic logic in computer science*. Wiley&Sons, 1993.
- [5] Timothy J. Norman, Carles Sierra, and Nick R. Jennings. Right and commitment in multi-agent agreements. In *In Proc. ICMAS-98*, pp. 222–229, 1998.

時刻 t	観測 (SO_t)	想起 (SR_t)
00320	Bi.patient(jo) @ i	
00331	[ans,tom,jo:bt] @ tom	○ Oi,tom.[ans,tom,jo:bt]
00332	Oi,tom.[ans,tom,jo:bt] @ i	○ Di.(jo:bt)
00333	Di.(jo:bt) @ i	○ Di.role(doctor) ● Di.(jo:bt)
00416	Ri.role(doctor) @ i	○ Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00417	Ri/doctor.[ask,db,jo:bt] @ i	○ Di/doctor.[ask,db,jo:bt] ● Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00418	Di/doctor.[ask,db,jo:bt] @ i	○ Ii/doctor.[ask,db,jo:bt] ● Di/doctor.[ask,db,jo:bt] ● Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00419	Ii/doctor.[ask,db,jo:bt] @ i	○ [ask,db,jo:bt] @ i/doctor ● Ii/doctor.[ask,db,jo:bt] ● Di/doctor.[ask,db,jo:bt] ● Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00423	Bi/doctor.(jo:bt>a) @ db	○ Ii/doctor.[ans,tom,jo:bt>a] ● Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00424	Ii/doctor.[ans,tom,jo:bt>a] @ i	○ [ans,tom,jo:bt>a] @ i/doctor ● Ii/doctor.[ans,tom,jo:bt>a] ● Ri/doctor.[ask,db,jo:bt] ● Di.(jo:bt)
00425	[ans,tom,jo:bt>a] @ i/doctor	

図 2 シミュレーション

- [6] Ichiro Osawa. A computational model of an intelligent agent for natural language dialogue systems. In *Proc. of PRICAI'90*, pp. 209–214, 1990.
- [7] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a bdi-architecture. Technical Report 14, AAIL, 1991.
- [8] Gerd Wagner. Towards agent-oriented information systems. Technical report, Freie University Berlin, 1999.
- [9] 大澤一郎. 人間と対話する知能エージェントのモデル. コンピュータ・ソフトウェア, Vol. 13, No. 2, pp. 35–44, 1996.

BDI Logic の sequent calculus による演繹体系

新出尚之 高田司郎

In this paper, we give a sound and complete deduction system for BDI logic whose base is restricted to propositional CTL, using Gentzen's sequent calculus. In this system we can construct a proof of a formula using Wang's algorithm with some extension. In this way, we can check the provability of a formula straightforwardly, and in addition, can give an easily understandable proof of the formula.

1 はじめに

BDI Logic [2] は, branching time temporal logic の一種である CTL*[1] に, BEL(信念), DESIRE(願望), INTEND(意図) といった心的状態を表す様相を導入して拡張した時相述語論理であり, マルチエージェント環境を含む合理的エージェントの仕様記述[2][5]などに用いられている.

BDI Logic を基礎概念においた合理的エージェントの実現手法として, BDI アーキテクチャ [4][6] が提案されている. しかし, 現状の BDI アーキテクチャは, 動作の基本となるループや心的状態の更新が手続き的に行われているなど, 論理体系との整合性が不十分な箇所が多いため, 仕様の自動検証が行いにくい, 論理的記述力への制約が多いなどの問題点がある.

我々はそれを解決する方法として, BDI アーキテクチャの Logic based な実現を検討している. そのためには, BDI Logic の公理化や, 一階述語論理の導出原理に相当するものの構築が必要となる. しかし, 現状では BDI Logic の公理化は十分とはいえず, ベースを命題論理の CTL に制限した BDI Logic の subset に対する健全で完全な演繹体系が Rao らによって与えられている [3] 程度であるが, これは Hilbert style のもので, 論理式の証明を見通しよく行うには向いておらず, BDI logic で書かれた仕様の論理的検証などにも向いていない.

本論文では, Rao らと同じく命題論理の CTL をベースとする BDI Logic の subset について, Gentzen の sequent calculus [7] による健全で完全な演繹体系を示す. これは, 彼らを与えた tableau 法による BDI logic の論理式の充足可能性の判定アルゴリズム [3] に, ほぼそのまま対応するものとなっている. また, この体系では, 命題論理の証明手続きとして知られる Wang のアルゴリズムを拡張したものを用いて, 論理式の証明可能性のチェックを見通しよく行え, 証明も人間にとって直感的に理解しやすいものが得られる.

2 論理式の定義

命題記号の集合 P をあらかじめ決めておく. 命題論理の CTL をベースとする BDI Logic の論理式は, 以下のように定義される.

- $p \in P$ ならば p は論理式 (これを原子命題という)
- ϕ と ψ が論理式ならば $\phi \vee \psi$, $\neg\phi$, $AX \phi$, $A(\phi U \psi)$, $E(\phi U \psi)$, $BEL(\phi)$, $DESIRE(\phi)$, $INTEND(\phi)$

Deduction Systems for BDI Logics Using Sequent Calculus.

NIDE Naoyuki, 奈良女子大学理学部情報科学科. Nara Women's University.

Shiro Takata, ATR メディア情報科学研究所. ATR Media Information Science Lab.

もそれぞれ論理式

必要に応じて括弧で曖昧さを除去. $true$ は $q \vee \neg q$ の略記 (但し q は適当に選んで固定した P の要素), $false$, $\phi \wedge \psi$, $\phi \supset \psi$, $\phi \Leftrightarrow \psi$ はそれぞれ $\neg true$, $\neg((\neg\phi) \vee (\neg\psi))$, $(\neg\phi) \vee \psi$, $(\phi \supset \psi) \wedge (\psi \supset \phi)$ の略記, また $EX\phi$, $AF\phi$, $EF\phi$, $AG\phi$, $EG\phi$ はそれぞれ $\neg AX\neg\phi$, $A(true \cup \phi)$, $E(true \cup \phi)$, $\neg EF\neg\phi$, $\neg AF\neg\phi$ の略記. 優先順位は, 単項論理記号 ($\neg$, AX など), $\wedge$, $\vee$, $\supset$ の順とし, 冗長な括弧は適宜省く.

3 意味論

3.1 BDI_{CTL}^K -structure

以下のものを定めておく.

- 可能世界の集合 $W (\neq \emptyset)$
- W の各要素 w に対して, state の集合 $St_w (\neq \emptyset)$ とその上の関係 $R_w \subset St_w \times St_w$ (但し, R_w は total であること. すなわち, 全ての $t \in St_w$ に対し, ある $t' \in St_w$ が存在して $t R_w t'$ を満たさねばならない)
- W の各要素 w と St_w の要素 t の組に対し, 命題記号への真偽値割り当て $L(w, t) \subseteq P$
- $W, \bigcup_{w \in W} St_w, W$ 上の 3 項関係 $\mathcal{B}, \mathcal{D}, \mathcal{I} \subset W \times \bigcup_{w \in W} St_w \times W$. 但し, 条件
 - $(w, t, w') \in \mathcal{B}$ かつ $t \in St_w$ ならば $t \in St_{w'}$
 - $(w, t, w') \in \mathcal{D}$ かつ $t \in St_w$ ならば $t \in St_{w'}$
 - $(w, t, w') \in \mathcal{I}$ かつ $t \in St_w$ ならば $t \in St_{w'}$
 が満たされねばならない^{†1}

このとき, tuple $M = \langle W, \{St_w \mid w \in W\}, \{R_w \mid w \in W\}, L, \mathcal{B}, \mathcal{D}, \mathcal{I} \rangle$ を BDI_{CTL}^K -structure と呼ぶ. また, それがさらに条件

- (B-D) 任意の $w \in W$ に対し, $(w, t, w') \in \mathcal{B}$ を満たす $w' \in W$ が存在する
- (B-4) 任意の $w, w', w'' \in W$ に対し, $(w, t, w') \in \mathcal{B}$ かつ $(w', t, w'') \in \mathcal{B}$ ならば $(w, t, w'') \in \mathcal{B}$
- (B-5) 任意の $w, w', w'' \in W$ に対し, $(w', t, w) \in \mathcal{B}$ かつ $(w', t, w'') \in \mathcal{B}$ ならば $(w, t, w'') \in \mathcal{B}$
- (D-D) 任意の $w \in W$ に対し, $(w, t, w') \in \mathcal{D}$ を

満たす $w' \in W$ が存在する

- (I-D) 任意の $w \in W$ に対し, $(w, t, w') \in \mathcal{I}$ を満たす $w' \in W$ が存在する

を全て満たせば $(B^{KD45}D^{KD}I^{KD})_{CTL}$ -structure と呼ぶ.

直感的には, state が一般の時相論理での「点時刻」, あるいは一般の様相論理の可能世界意味論での「可能世界」にあたり, R_w が時間の前後関係, $\mathcal{B}, \mathcal{D}, \mathcal{I}$ が様相としての信念・願望・意図を表す可視関係である.

3.2 論理式の解釈と恒真性

$w \in W$ に対し, St_w に属する state の無限列 $(t_0, t_1, \dots)$ で, 条件 $t_0 R_w t_1, t_1 R_w t_2, \dots$ を満たすものを, t_0 で始まる w 上の path という.

BDI_{CTL}^K -structure $M = \langle W, \{St_w \mid w \in W\}, \{R_w \mid w \in W\}, L, \mathcal{B}, \mathcal{D}, \mathcal{I} \rangle$ と世界 $w \in W$, state $t \in St_w$ に対し, 論理式 ϕ が真である $((M, w, t) \models \phi$ と書く) ための条件を, 以下のように定義する.

- ϕ が primitive proposition である ($\phi \in P$) 場合, $(M, w, t) \models \phi$ iff $\phi \in L(w, t)$
- $(M, w, t) \models \neg\phi$ iff $(M, w, t) \not\models \phi$
- $(M, w, t) \models \phi \vee \psi$ iff $(M, w, t) \models \phi$ or $(M, w, t) \models \psi$
- $(M, w, t) \models AX\phi$ iff for all t' s.t. $t R_w t'$, $(M, w, t') \models \phi$
- $(M, w, t_0) \models A(\phi \cup \psi)$ iff t_0 で始まる w 上の任意の path $(t_0, t_1, \dots)$ に対し, その中のある state t_i があって $(M, w, t_i) \models \psi$ かつ for all j s.t. $0 \leq j < i$, $(M, w, t_j) \models \phi$
- $(M, w, t_0) \models E(\phi \cup \psi)$ iff t_0 で始まる w 上のある path $(t_0, t_1, \dots)$ とその中のある state t_i があって, $(M, w, t_i) \models \psi$ かつ for all j s.t. $0 \leq j < i$, $(M, w, t_j) \models \phi$
- $(M, w, t) \models BEL(\phi)$ iff for all w' s.t. $(w, t, w') \in \mathcal{B}$, $(M, w', t) \models \phi$
- $(M, w, t) \models DESIRE(\phi)$ iff for all w' s.t. $(w, t, w') \in \mathcal{D}$, $(M, w', t) \models \phi$
- $(M, w, t) \models INTEND(\phi)$ iff for all w' s.t. $(w, t, w') \in \mathcal{I}$, $(M, w', t) \models \phi$

論理式 ϕ が, 全ての BDI_{CTL}^K -structure M の全ての世界 w , state t に対し $(M, w, t) \models \phi$ を満たすと

^{†1} Rao らの論文[2][3]ではこの条件を明示的に要請してはいないが, BEL などの様相記号の意味の定義が整合的であるためには明らかに必要であろう.

き, ϕ は $\text{BDI}_{CTL}^K\text{-valid}$ であるといい, $\text{BDI}_{CTL}^K \models \phi$ と書く. また, 全ての $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}\text{-structure } M$ について同様の条件を満たすとき, ϕ は $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}\text{-valid}$ であるといい, $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL} \models \phi$ と書く (いずれも, 誤解のおそれのない場合単に $\models \phi$ と書き, 単に valid であるということがある).

4 演繹体系

本章では演繹体系 $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}$ および BDI_{CTL}^K を導入する. なお本稿では, sequent の $\rightarrow$ の両側は論理式の multi set とする (従って exchange の推論規則は必要ない). また本稿では, 文中で sequent の範囲を明示するために, sequent 全体を $[]$ でくくることがある.

以下, ギリシア大文字は論理式の multi set, ギリシア小文字は単一の論理式を表す. また, 論理式の multi set Γ と単項論理演算子 K に対し, $K(\Gamma)$ で Γ の各論理式に K を付加して得られる multi set を表す.

4.1 推論規則

$(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}$ の演繹体系は, 表 1 の推論規則のうち BEL-K, DESIRE-K, INTEND-K 以外の全て (これらを含めてもよいが冗長), また BDI_{CTL}^K の演繹体系は同表のうち BEL-KD45, DESIRE-KD, INTEND-KD 以外の全てを推論規則として持つ.

これらの規則は, 実は Rao らの BDI logic の決定アルゴリズム [3] の pseudo tableau の構築の各 step に対応している. 例えば規則 BEL-KD45 は, [3] の 6.2 節, pseudo- $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}\text{-tableau}$ の構築における (d) に相当する.

この体系では, 前提が結論より大きくなる規則が存在するので, Wang のアルゴリズムをそのまま適用はできない. しかし, 6 章で述べるように, Weak の形を制限することによって, 証明図が必ず有限で止まるかループし, 無限に伸びないようにすることができる. 従って証明可能性の判定は有限の手間で行える.

4.2 証明可能性の定義

ここでは, BDI_{CTL}^K および $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}$ の演繹体系での, sequent の証明可能性の定義を与える.

定義 4.1 S を sequent とする. 各ノードが 1 つの sequent をラベルとして持ち, 根のラベルが S であり, かつ葉以外の任意のノード N が条件

N の子ノードが $N_1, \dots, N_i$ であり, N および $N_1, \dots, N_i$ のラベルが $S', S_1, \dots, S_i$ であるならば, $\frac{S_1 \dots S_i}{S'}$ という推論規則が存在する

を満たす木を, S の推論木という.

以下では, $\rho = A(\phi \cup \psi)$ の場合は $\text{AX} \rho$ を, また $\rho = E(\phi \cup \psi)$ の場合は $\text{EX} \rho$ を, $\text{AEX} \rho$ と表記する.

定義 4.2 S の推論木 T の葉でないノード N_0 から, その子孫の葉ノード L までの経路 $N_0, N_1, \dots, N_n (= L)$ が, 以下の条件を全て満たすとき, この経路を **導出のループ**という.

1. N_0 と L は等しいラベルを持つ
2. $N_1, \dots, N_{n-1}$ の中には L と等しいラベルを持つものはない

定義 4.3 ある論理式 $\rho = A(\phi \cup \psi)$ あるいは $\rho = E(\phi \cup \psi)$ が存在して, 導出のループ $N_0, N_1, \dots, N_n$ とその各ノードのラベル $S_0, S_1, \dots, S_n (= S_0)$ が以下の条件を全て満たすとき, これを **終局性論理式 ρ を持つ導出のループ**という.

1. $S_i = [\rho, \Gamma \rightarrow \Delta]$ かつ $S_{i+1} = [\phi, \text{AEX} \rho, \Gamma \rightarrow \Delta]$ を満たす整数 i ($0 \leq i < n$) と Γ, Δ が存在する. また, $S_i = [\rho, \Gamma \rightarrow \Delta]$ かつ $S_{i+1} = [\psi, \Gamma \rightarrow \Delta]$ を満たす整数 i ($0 \leq i < n$) と Γ, Δ は存在しない. (大まかには, S_0 から S_n までの過程のどこかで, ρ に対する AU 左か EU 左が適用されており, かつ, 導出のループはその前提のうち右へ行く経路)
2. $S_i = [\Gamma, \Gamma' \rightarrow \Delta, \Delta']$ および $S_{i+1} = [\Gamma \rightarrow \Delta]$ を満たす全ての整数 i ($0 \leq i < n$) と $\Gamma, \Gamma', \Delta, \Delta'$ に対し, Γ' に含まれる全ての論理式は

- 原子命題
- トップレベルのオペレータが BEL, DESIRE, INTEND のいずれかである論理式
- Γ に含まれる論理式

のいずれかである. また, Δ' に含まれる全ての論理式は

- 原子命題
- トップレベルのオペレータが BEL, DESIRE, INTEND のいずれかである論理式

表 1 $(B^{KD45}D^{KD}I^{KD})_{CTL}$, BDI_{CTL}^K の推論規則

$\frac{\Gamma \rightarrow \phi, \Delta}{\neg \phi, \Gamma \rightarrow \Delta} (\neg\text{左})$	$\frac{\phi, \Gamma \rightarrow \Delta \quad \psi, \Gamma \rightarrow \Delta}{\phi \vee \psi, \Gamma \rightarrow \Delta} (\vee\text{左})$	$\frac{\phi, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \neg \phi, \Delta} (\neg\text{右})$	$\frac{\Gamma \rightarrow \phi, \psi, \Delta}{\Gamma \rightarrow \phi \vee \psi, \Delta} (\vee\text{右})$	$\frac{\Gamma \rightarrow \Delta}{\Gamma, \Gamma' \rightarrow \Delta, \Delta'} (\text{Weak})$
$\frac{\psi, \Gamma \rightarrow \Delta \quad \phi, AX A(\phi \cup \psi), \Gamma \rightarrow \Delta}{A(\phi \cup \psi), \Gamma \rightarrow \Delta} (\text{AU 左})$	$\frac{\Gamma \rightarrow \psi, \phi, \Delta \quad \Gamma \rightarrow \psi, AX A(\phi \cup \psi), \Delta}{\Gamma \rightarrow A(\phi \cup \psi), \Delta} (\text{AU 右})$	$\frac{\psi, \Gamma \rightarrow \Delta \quad \phi, EX E(\phi \cup \psi), \Gamma \rightarrow \Delta}{E(\phi \cup \psi), \Gamma \rightarrow \Delta} (\text{EU 左})$	$\frac{\Gamma \rightarrow \psi, \phi, \Delta \quad \Gamma \rightarrow \psi, EX E(\phi \cup \psi), \Delta}{\Gamma \rightarrow E(\phi \cup \psi), \Delta} (\text{EU 右})$	
$\frac{\Gamma \rightarrow \Theta}{AX(\Gamma) \rightarrow AX(\Theta)} (\text{AX-KD})$	$\frac{\Gamma \rightarrow \phi}{BEL(\Gamma) \rightarrow BEL(\phi)} (\text{BEL-K})$	$\frac{\Gamma, BEL(\Gamma) \rightarrow BEL(\Delta), \Theta, BEL(\Theta)}{BEL(\Gamma) \rightarrow BEL(\Delta), BEL(\Theta)} (\text{BEL-KD45})$		
$\frac{\Gamma \rightarrow \phi}{DESIRE(\Gamma) \rightarrow DESIRE(\phi)} (\text{DESIRE-K})$	$\frac{\Gamma \rightarrow \Theta}{DESIRE(\Gamma) \rightarrow DESIRE(\Theta)} (\text{DESIRE-KD})$			
$\frac{\Gamma \rightarrow \phi}{INTEND(\Gamma) \rightarrow INTEND(\phi)} (\text{INTEND-K})$	$\frac{\Gamma \rightarrow \Theta}{INTEND(\Gamma) \rightarrow INTEND(\Theta)} (\text{INTEND-KD})$			

(但し, Θ は高々 1 つの論理式とする)

- Δ に含まれる論理式

のいずれかである.

(大まかには, S_0 から S_n までの過程で Weak を適用して, AX, AU, EU をトップレベルに持つ論理式が消されることはない)

定義 4.4 sequent S が**証明可能**であるとは, 以下の条件を全て満たす S の推論木 T が存在することをいう. ここで *initial sequent* とは, $[\Gamma, \phi \rightarrow \Delta, \phi]$ という形の sequent である.

1. T の全ての葉は, initial sequent をラベルとして持つか, あるいは導出のループの終点である
2. T の全ての導出のループを合わせたグラフ (導出のループに含まれるノードと枝全てからなるグラフ) の各連結成分 T' (これは T の部分木になる) に対し, ある論理式 ρ があって, T' に含まれる T の導出のループは全て, 終局性論理式 ρ を持つ

BDI_{CTL}^K および $(B^{KD45}D^{KD}I^{KD})_{CTL}$ の演繹体系において, $[\rightarrow \phi]$ という sequent が証明可能であるとき, 論理式 ϕ が証明可能であるという.

4.3 例

図 1 は, $(B^{KD45}D^{KD}I^{KD})_{CTL}$ における論理式 $A(p \cup BEL(q)) \supset A(p \cup BEL(BEL(q)))$ の証明である (スペースの都合で図の一部を持ち上げている). この論理式は $\neg A(p \cup BEL(q)) \vee A(p \cup BEL(BEL(q)))$ の略記であるため, 証明の一番下はそれに $\rightarrow$ を前置したのになっている. なお, 下から 3 段目の $A(p \cup BEL(q)) \rightarrow A(p \cup BEL(BEL(q)))$ から一番右

の葉までが, 終局性論理式 $A(p \cup BEL(q))$ を持つ導出のループになっていることに注意.

この証明の下から 3 段目以上は, 直感的には「 $BEL(q)$ ならば $A(p \cup BEL(BEL(q)))$ 」と, 「『 $A(p \cup BEL(q))$ ならば $A(p \cup BEL(BEL(q)))$ 』がある時刻で成り立つならばその直前の時刻でも成り立つ」を示すことによって, 数学的帰納法に近い感覚で「全ての時刻で『 $A(p \cup BEL(q))$ ならば $A(p \cup BEL(BEL(q)))$ 』が成り立つ」を示していることになる. このように, 得られる証明は, Hilbert style の演繹体系に比べ, 直感的な理解しやすさで優るものになる.

また, 全く同様にして, 例えば $A(INTEND(AF \phi) \cup BEL(\phi)) \supset A(INTEND(AF \phi) \cup BEL(\phi) \vee \neg BEL(EF \phi))$ の証明もできる. これは, コミットメント戦略[2]の blind commitment が single-minded commitment より強いことの証明になっている.

5 健全性

本章では $(B^{KD45}D^{KD}I^{KD})_{CTL}$ の健全性を示す. BDI_{CTL}^K の健全性もほぼ同様に示せる.

sequent $S = [\phi_1, \phi_2, \dots, \phi_n \rightarrow \psi_1, \psi_2, \dots, \psi_m] (n \geq 0 \leq m)$ に対し, 論理式 $\neg \phi_1 \vee \dots \vee \neg \phi_n \vee \psi_1 \vee \dots \vee \psi_m$ (但し $n = m = 0$ のときは *false*) を, S に対応する論理式と呼ぶ. S に対応する論理式が (M, w, t) で真のとき, (M, w, t) で S が真であるといい, $(M, w, t) \models S$ と書く. 特に, sequent S が valid であるとは, S に対応する論理式が valid であることと定義する.

補題 5.1 導出のループの中には, 空 sequent $[\rightarrow]$ は現れない.

図 1 $(B^{KD45}D^{KD}I^{KD})_{CTL}$ での証明の例

$$\begin{array}{c}
\frac{A(p \cup BEL(q)) \rightarrow A(p \cup BEL(BEL(q)))}{AX A(p \cup BEL(q)) \rightarrow AX A(p \cup BEL(BEL(q)))} (AX-KD) \\
\frac{p, AX A(p \cup BEL(q)) \rightarrow BEL(BEL(q)), p}{p, AX A(p \cup BEL(q)) \rightarrow BEL(BEL(q)), AX A(p \cup BEL(BEL(q)))} (Weak) \\
\frac{p, AX A(p \cup BEL(q)) \rightarrow BEL(BEL(q)), p}{p, AX A(p \cup BEL(q)) \rightarrow A(p \cup BEL(BEL(q)))} (AU 右) \\
\frac{q, BEL(q) \rightarrow BEL(q), BEL(BEL(q))}{BEL(q) \rightarrow BEL(BEL(q))} (Weak) \quad \frac{q, BEL(q) \rightarrow BEL(q), BEL(BEL(q))}{BEL(q) \rightarrow BEL(BEL(q))} (Weak) \\
\frac{BEL(q) \rightarrow BEL(BEL(q)), p}{BEL(q) \rightarrow BEL(BEL(q)), AX A(p \cup BEL(BEL(q)))} (Weak) \quad \frac{BEL(q) \rightarrow BEL(BEL(q))}{BEL(q) \rightarrow A(p \cup BEL(BEL(q)))} (AU 左) \\
\frac{A(p \cup BEL(q)) \rightarrow A(p \cup BEL(BEL(q)))}{\rightarrow \neg A(p \cup BEL(q)), A(p \cup BEL(BEL(q)))} (\neg 右) \\
\frac{\rightarrow \neg A(p \cup BEL(q)), A(p \cup BEL(BEL(q)))}{\rightarrow \neg A(p \cup BEL(q)) \vee A(p \cup BEL(BEL(q)))} (\vee 右)
\end{array}$$

証明 定義から、導出のループは長さ 2 以上でなければならないが、結論を空、前提を非空にできる推論規則は存在しない。□

補題 5.2 導出のループの中では、推論規則 AX-KD か BEL-KD45 のいずれかが少なくとも 1 回適用されており、かつ、その両方は使われていない。また、DESIRE-KD、INTEND-KD は使われていない。

証明 導出のループの中では、前提が結論より小さくない推論規則が使われていなければならない。従って、AU/EU 左・右のいずれかか、または BEL-KD45 が使われている。

もし AU/EU 左・右が使われているならば、そこで AX より内側にない AU(または EU) を 1 つ減らしている (ここで「減る」とは、 S_i に対して S_{i+1} で、すなわち結論に対して前提で減っていることを意味する。以下も同様)。よって、ループであるためにはそれをどこかで増やさなければならないが、それができる規則は AX-KD か BEL-KD45 だけである。従って AX-KD^{†2}あるいは BEL-KD45 が使われている。

仮に、その両方が使われているとする。BEL-KD45 の結論は補題 5.1 より空ではないので、それに現れる論理式のうち、BEL を最も多く持つものの全ての集合を BEL(≡) とする。すると、この先 Weak が適用されるまでは、「sequent に現れる論理式のうち BEL を最も多く持つものの全ての集合は BEL(≡) である」という

事実は変わらない。

しかし、AX-KD を適用するには、Weak で BEL(≡) を消さねばならず、この時、sequent に現れる各論理式が持つ BEL の個数の最大は減る。そして、それを増やせる規則は存在しない。従って、BEL-KD45 が使われているならば AX-KD は使われていない。

また、DESIRE-KD が使われているとすると、sequent に現れる各論理式が持つ DESIRE の個数の最大がそこで減るが、これを増やす規則はない。従って、DESIRE-KD は使われていない。INTEND-KD も同様である。□

補題 5.3 終局性論理式 ρ を持つ導出のループの中では、推論規則 AX-KD が少なくとも 1 回適用されており、かつ BEL-KD45、DESIRE-KD、INTEND-KD は使われていない。また、AX-KD のある適用から次の適用までの間には、 ρ に対する AU/EU 左が少なくとも 1 回適用されている。

証明 終局性論理式 ρ を持つ導出のループの中には、AU/EU 左が適用されて ρ が $\wedge EX \rho$ になっている箇所がある。Weak の形の制限 (定義 4.3 の 2) から、 ρ や $\wedge EX \rho$ は Weak では消せないで、sequent に現れる論理式のトップレベルを BEL ばかりにすることはできず、BEL-KD45 は使えない。従って補題 5.2 から、AX-KD が使われており、DESIRE-KD、INTEND-KD は使われていない。

また、同じ理由から、 ρ は、AU/EU 左の適用で $\wedge EX \rho$ に変化しては、AX-KD の適用で ρ に戻すことを繰り返す。従って AX-KD の適用から適用までの間には、 ρ に対する AU/EU 左が必ず適用されている。□

補題 5.4 AX-KD、BEL-KD45、INTEND-KD、

^{†2} 但しこの証明では、ループを 1 周して戻ってきた時の ρ は、AU/EU 左の適用で得られた $\wedge EX \rho$ の $\wedge EX$ を AX-KD で除去して得たものであるとは必ずしも保証していないことに注意。他の論理式の部分論理式であった ρ が分解されて出てきた可能性を排除していないからである。

DESIRE-KD 以外のある推論規則の前提が $S_1, \dots, S_n$, 結論が S であり, かつ $(M, w, t) \not\models S$ となる (M, w, t) が存在するならば, $(M, w, t) \not\models S_i$ となる $i (1 \leq i \leq n)$ が存在する.

また, AX-KD の前提が S_1 , 結論が S であり, かつ $(M, w, t) \not\models S$ となる (M, w, t) が存在するならば, $t R_w t'$ かつ $(M, w, t') \not\models S_1$ となる t' が存在する.

証明 論理式の解釈の定義より直ちに従う. $\square$

補題 5.5 ρ が $A(\phi \cup \psi)$ か $E(\phi \cup \psi)$ のいずれかであるとする. $(M, w, t) \not\models [\rho, \Gamma \rightarrow \Delta]$ かつ $(M, w, t) \models [\psi, \Gamma \rightarrow \Delta]$ ならば, $(M, w, t) \models \rho$ かつ $(M, w, t) \models \psi$ である.

証明 論理式の解釈の定義より直ちに従う. $\square$

補題 5.6 終局性論理式 $\rho = E(\phi \cup \psi)$ を持つ導出のループの中で, ノード N と N' の間で AX-KD が適用されており (すなわち, N のラベル S が AX-KD の結論で, N' のラベル S' がその前提), しかも $(M, w, t) \not\models S$ とする. すると, $t R_w t'$ なる任意の t' に対し, $(M, w, t') \not\models S'$ iff $(M, w, t') \models \rho$ となる.

証明 ある Γ が存在して $S = [AX(\Gamma) \rightarrow AX\neg\rho]$, $S' = [\Gamma \rightarrow \neg\rho]$ と書けることから, 論理式の解釈の定義より直ちに従う. $\square$

定理 5.7 体系 $(B^{KD45}D^{KD}I^{KD})_{CTL}$ は健全である.

証明 sequent S の推論木 T の全ての導出のループを合わせたグラフ G の連結成分の個数 j に関する帰納法で示す. $j = 0$ の場合は trivial なので, $j < k (k > 0)$ の場合を仮定して $j = k$ の場合を示す.

S が valid でないとする. 全ての推論規則は, 前提が全て valid ならば結論も valid であるので, 結論が valid でないなら前提のどれかが valid でない. 従って T の根から子へ, ラベルが valid でないノードを選んでたどっていくことができる. するといつかは, G のある連結成分 T' の根 R に着く. なぜなら, そうでないと導出のループの終点でない葉に着いてしまい, 矛盾するからである.

T' に含まれる T の導出のループが共通に持つ終局性論理式を ρ とする. $\rho = A(\phi \cup \psi)$ または $\rho = E(\phi \cup \psi)$ である.

R からさらに子へ, ラベルが valid でないノードを選んでたどっていく. このとき T' の外に出ることは

ない. なぜなら, R の子孫のうち T' に属さないノードのラベルは, 帰納法の仮定から valid だからである.

するといつかは, あるノード N' とその子ノード N_0 の間で, ρ に対する AU 左か EU 左が適用されている箇所に着く.

N' , N_0 のラベルをそれぞれ S' , S_0 とすると, S' が valid でないことから, ある (M, w, t_0) が存在して $(M, w, t_0) \not\models S'$ である. また, ここで適用される AU 左か EU 左の前提のうち左は, 定義 4.3 の 1 から T' の外であり, 従って valid である. よって補題 5.5 から $(M, w, t_0) \models \rho$ かつ $(M, w, t_0) \models \psi$ である. また, $(M, w, t_0) \models S_0$ も成り立つ.

補題 5.3 より, N_0 からその子ノードへの過程で使われている推論規則は BEL-KD45, DESIRE-KD, INTEND-KD 以外のいずれかである. 補題 5.4 より, ここで使われている推論規則が AX-KD 以外であれば, $(M, w, t_0) \not\models S_1$ を満たすラベル S_1 を持つ N_0 の子ノード N_1 があり, また, AX-KD であれば, N_0 の唯一の子ノード N_1 のラベル S_1 に対し, $t_0 R_w t_1$ かつ $(M, w, t_1) \not\models S_1$ となる t_1 が存在する. いずれの場合も N_1 は T' の中にある.

以下, N_0 から N_1 への過程と同様にして, N_1 からそのある子ノード N_2 へ, さらに子ノード N_3 へ…とたどることを繰り返す, T の葉 (T' の葉でもある) にたどり着いたら, それと同じラベルを持つ T' の (葉でない) ノードに戻ってさらに続けていくことにより, ノードの無限列 $N_0, N_1, N_2, \dots$ とそのラベル $S_0, S_1, S_2, \dots$, および w 上の path $t_0, t_1, \dots$ を, $(M, w, t_{applAX(i)}) \not\models S_i$ となるようにとることができる. ここで $applAX(i)$ とは, S_0 から S_i までの間に AX-KD が適用された回数である.

補題 5.3 より, S_0 の $\rightarrow$ の左にある $AEX\rho$ は, AX-KD を適用されて ρ に変わっては, AU 左か EU 左の適用によって $AEX\rho$ に戻ることを繰り返す. N' から N_0 への過程における議論と同様にして, 任意の自然数 ℓ に対し, $(M, w, t_\ell) \models \rho$ かつ $(M, w, t_\ell) \not\models \psi$ となる. すなわち, path $t_0, t_1, \dots$ において, ずっと ρ が真かつ ψ が偽である. $\rho = A(\phi \cup \psi)$ であれば, これは論理式の解釈の定義に反する.

一方, $\rho = E(\phi \cup \psi)$ であれば, path $t_0, t_1, \dots$ が

上記のようにして取った path であることと, path $t_0, t_1, \dots$ が任意の自然数 ℓ に対し $(M, w, t_\ell) \models \rho$ を満たすこととは同値であることが, 補題 5.6 を繰り返し適用することにより示せる. しかも, 上記のようにして取った path では, 任意の自然数 ℓ に対し $(M, w, t_\ell) \models \psi$ となるので, t_0 から始まる pathのうち, ρ が偽にならないうちに ψ が真となるものは存在しない. 従ってやはり論理式の解釈の定義に反する.

以上から, S は valid である. $\square$

6 完全性

本章では $(B^{KD45}D^{KD}I^{KD})_{CTL}$ の完全性を示す. BDI_{CTL}^K の完全性も同様に示せる.

まず 6.1 節で, 証明可能でない sequent S を与えたとき, ある $(B^{KD45}D^{KD}I^{KD})_{CTL}$ -structure を構築するアルゴリズムを与える. 続いて 6.2 節で, この $(B^{KD45}D^{KD}I^{KD})_{CTL}$ -structure のある世界において S が偽となることを示す.

6.1 $(B^{KD45}D^{KD}I^{KD})_{CTL}$ -structure の構築アルゴリズム

S を, 証明可能でない sequent とする. 以下の作業を順に行う.

6.1.1 有限木の構築

まず $S = S_0$ とおき, S_0 に次の‘手順A’を適用する.

手順A begin

S_0 に Wang のアルゴリズムを準用し, $\neg/\vee/AU/EU$ 左・右を可能な限り適用する. その後, もし各 sequent の $\rightarrow$ の左同士か右同士に同じ論理式が複数回現れていれば, Weak でそれを 1 つに減らす.

手順A end

この結果得られる sequent のうち, 証明可能でないもの全て (1 つ以上有限個) の集合を $\{S'_1, S'_2, \dots\}$ とおき, そのうち任意の 1 つを S' と置く. S' は, $\rightarrow$ の左右に原子命題とトップレベルが AX, BEL, DESIRE, INTEND の論理式だけを持ち, しかも $\rightarrow$ の左同士や右同士に同じ論理式が複数回現れない.

ただ 1 つの要素 w_0 を持つ集合 W と, ただ 1 つの要素 t_0 を持つ集合 St を作る. また, ペア (w_0, t_0) を唯一のノードとする木 T を作り, このノードに第 1 ラ

ベルとして sequent S , 第 2 ラベルとして sequent S' を貼る.

次いで, $w = w_0, t = t_0$ と置いて, S' とペア (w, t) に対し, 以下の‘手順B’ (図 2) を (再帰的に) 実行する.

手順B begin

S' の $\rightarrow$ の左右が原子命題ばかりであるなら, 何もしない. また, T の根 (w_0, t_0) からノード (w, t) までの経路のどこか (但し (w, t) は含まない) に, 第 2 ラベル S' を持つノードがあるならば, ノード (w, t) に「ループ」というマークを付け, 後は何もしない.

いずれでもなければ, S' に様相論理版の Wang のアルゴリズムを準用する. すなわち, S' に対し以下のいずれかを行って得られる sequent 全ての有限集合を $\{S''_1, \dots, S''_k\}$ とする.

1. Weak で, $\rightarrow$ の左ではトップレベルが AX の論理式以外を全て消す. $\rightarrow$ の右では, トップレベルが AX の論理式があれば, そのうち 1 つを残して他を消し, そうでなければ全て消す.
次いで AX-KD を適用する.
2. Weak で, $\rightarrow$ の左右それぞれについて, トップレベルが BEL の論理式全てを残し, 他を消す.
続いて BEL-KD45 を適用する. その際, $\rightarrow$ の右が空なら Δ, Θ を空とおき, そうでなければ $\rightarrow$ の右のうち 1 つを Θ , それ以外を Δ と置く.
3. DESIRE, INTEND に対し, 1. と同様のことを行う.

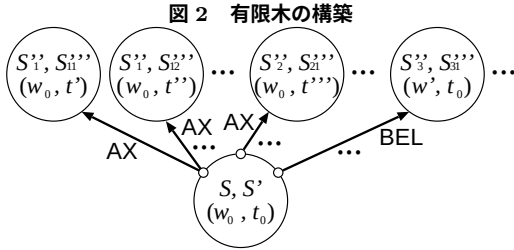
このとき, $S''_1, \dots, S''_k$ はいずれも証明可能でない. そこで, 各 S''_i ($1 \leq i \leq k$) に対し, $S_0 = S''_i$ と置いて‘手順A’を適用し, sequent $S'''_{i1}, S'''_{i2}, \dots$ を得る. $S'''_{i1}, S'''_{i2}, \dots$ はそれぞれ, $\rightarrow$ の左右に原子命題とトップレベルが AX, BEL, DESIRE, INTEND の論理式だけを持ち, $\rightarrow$ の左同士や右同士に同じ論理式が複数現れない, 証明可能でない sequent である.

ここで, 各 S'''_{ij} に対し, 以下を行う.

1. $S''' = S'''_{ij}$ と置く.
2. S' から S''_i を得る際に AX-KD, BEL-KD45, DESIRE-KD, INTEND-KD のどれを使ったかによって, 次のいずれかを行う.

(a) AX-KD を使った場合

St に新たな要素 t' を加え, T のノード



(w, t) に新たな子 (w, t') を作って、それに第 1 ラベルとして sequent S''_i , 第 2 ラベルとして sequent S''' を貼り、新たな枝に AX というタグを付ける
(直観的には、 t から t' に AX による可視関係の枝を張る)

(b) BEL-KD45 を使った場合

W に新たな要素 w' を加え、 T のノード (w, t) に新たな子 (w', t) を作って、それに第 1 ラベルとして sequent S''_i , 第 2 ラベルとして sequent S''' を貼り、新たな枝に BEL というタグを付ける
(直観的には、 w から BEL による可視関係で見える新たな可能世界を作る)

(c) DESIRE-KD, INTEND-KD を使った場合
DESIRE, INTEND に対し、2(b) と同様のことを行う

3. 次いで、今新設したノードを改めて (w, t) と置き、 S''' を改めて S' と置いて、‘手順B’を再帰的に呼び出す。

手順B end

どの段階においても、 S' は、 S に含まれる論理式の部分論理式 (その中に $A(\phi \cup \psi)$ がある場合は $AXA(\phi \cup \psi)$ を含み、また $E(\phi \cup \psi)$ がある場合は $EXE(\phi \cup \psi)$, $AX \neg E(\phi \cup \psi)$, $\neg E(\phi \cup \psi)$ を含む) のみからなり、しかも $\rightarrow$ の左同士や右同士に同じ論理式はない。従って S' の可能性は有限通りしかない。また、1 つの S' に対する S''_i の可能性も有限個しかない。従って ‘手順B’ は有限の手間で止まる。

このとき、 T は W の要素と St の要素のペアをノードとする有限木で、各ノードには sequent のラベルが 2 つ貼られており、各枝には AX, BEL, DESIRE,

INTEND のいずれかのタグが付いている。また、一部の葉には「ループ」というマークがつく。

6.1.2 枝刈り

T に対し、以下の作業を可能な限り繰り返す。

1. 「ループ」や「行き止まり」のマークが付いていないノードで、しかも次のいずれかの条件

- (a) 葉である
- (b) 子のノードに全て「行き止まり」マークが付いている

を満たすものがあれば、それに「行き止まり」マークを付ける

2. 「行き止まり」マークの付くノード N があり、 N と同じ親および同じ第 1 ラベルを持つノード $N' (\neq N)$ があれば、 N' を削除 (N' への枝、および N' 以下の部分木も含む)

木の有限性から、この操作は必ず停止し、その結果、同じ第 1 ラベルを持つ兄弟ノードは「行き止まり」マークの付くもののみ 1 個か、付かないもののみ 0 個以上のいずれかになる。また、この操作の後も T は木である。

6.1.3 ループ枝の融合

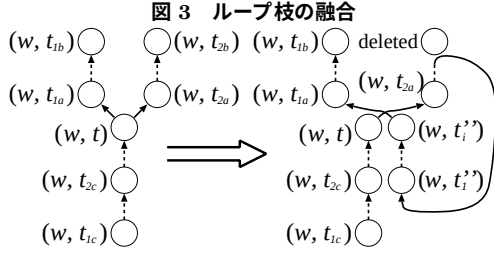
以下の全ての条件

- $(w, t_{1a}), (w, t_{2a})$ は同じ親 (w, t) および同じ第 1 ラベルを持つノードで、 $t_{1a} \neq t_{2a}$
- $(w, t_{1b}), (w, t_{2b})$ は「ループ」マークのある葉ノードで、 (w, t_{1b}) は (w, t_{1a}) の、 (w, t_{2b}) は (w, t_{2a}) の子孫
- (w, t_{1c}) は (w, t_{1b}) と、 (w, t_{2c}) は (w, t_{2b}) と同じ第 2 ラベルを持つノードで、 (w, t_{1c}) は (w, t_{2c}) の、 (w, t_{2c}) は (w, t) の先祖

を満たすノード $(w, t), (w, t_{1a}), (w, t_{2a}), (w, t_{1b}), (w, t_{2b}), (w, t_{1c}), (w, t_{2c})$ が存在する限り、以下の ‘手順C’ (図 3) を繰り返す。

手順C begin

まず、 (w, t_{2c}) から (w, t) までの経路のコピーを作る。すなわち、 (w, t_{2c}) から (w, t) までの経路のノードが $(w, t'_1), \dots, (w, t'_i)$ であるとする (ここで $t'_1 = t_{2c}, t'_i = t$) と、 St に新たな要素 $t''_1, \dots, t''_i$ を増やし、 $(w, t'_1), \dots, (w, t'_i)$ とそれぞれ同じ第 1・2 ラベルを持つノード $(w, t''_1), \dots, (w, t''_i)$ を作って (w, t'_1)



から (w, t_2'') へ, $\dots$, (w, t_{i-1}'') から (w, t_i'') へタグ AX を持つ枝を張る. また, ノード (w, t_1'') , $\dots$, (w, t_i'') に「コピー」のマークを付ける.

次いで, (w, t_1'') の第 1 ラベルのみ, (w, t_{2b}) のそれと同じに変更する. 但し (w, t_1'') に「ループ」のマークは付けない.

最後に, (w, t_{2b}) の親から (w, t_1'') へ, および (w, t_i'') から (w, t_{1a}) へタグ AX を持つ枝を張り, (w, t_{2b}) の親から (w, t_{2b}) へ, および (w, t) から (w, t_{1a}) への枝と, ノード (w, t_{2b}) を削除する.

手順C end

直観的にはこの操作は, タグ AX の枝からなるループ全てを含むグラフの各連結成分を, それと同等な枝を全て持つ単一のループに置き換える.

「手順C」1 回あたり, 条件を満たす箇所が 1 つ減るので, 繰り返しは必ず停止する. この操作の後 T は木であり, また, 1 つのノードから出る, タグ AX を持つ枝で, その先のノードの第 1 ラベルが同じものは高々 1 つになる. 特に, 「コピー」のマークを持つノードの子はただ 1 つである.

その後, 「コピー」のマークのある各ノード N に対し, その祖先で N と同じ第 2 ラベルを持ち「コピー」マークのないノード N' (必ず存在する) を選び, N の「コピー」のマークを外し, N' 以下の部分木 (但し, N の子と同じ第 1 ラベルを持つ N' の子と, それへの枝, およびそれ以下の部分木を除く) を, 必要なら W や St の要素を増やしつつ, N 以下へコピーする. このとき, コピーした部分木においても, ノード (w_1, t_1) からその子ノード (w_2, t_2) への枝のタグが AX であれば $w_1 = w_2$, そうでなければ $t_1 = t_2$ となるようにする. また, コピー元の部分木の中にさらに「コピー」のマークを持つノードがあれば, 先に再帰的に同様の

コピーを行う. この再帰のネストは, 高々「手順C」を行った回数なので, この操作は有限で停止する.

6.1.4 $(B^{KD45}D^{KD}I^{KD})_{CTL}$ -structure の構築

以下のような tuple $M = \langle W, \{St_w \mid w \in W\}, \{R_w \mid w \in W\}, L, B, D, I \rangle$ を作る.

- 各 $w \in W$ に対し, $St_w = \{t \in St \mid T \text{ がノード } (w, t) \text{ を持つ}\}$
- 各 $w \in W$ に対し, $R_w = \{(t, t') \mid T \text{ は } (w, t) \text{ から } (w, t') \text{ へのタグ AX の枝を持つ}\}$
- 各 $w \in W, t \in St_w$ に対し, $L(w, t)$ はノード (w, t) の第 2 ラベルである sequent の $\rightarrow$ の左に現れる原子命題全ての集合
- 各 $w, w' \in W, t \in St_w$ に対し, $B = \{(w, t, w') \mid T \text{ は } (w, t) \text{ から } (w', t) \text{ へのタグ BEL の枝を持つ}\}$
- D, I に対しては同様

次に, T の根からそれぞれの葉 L への経路のうち, 途中 (葉を除く) に葉と同じ第 2 ラベルを持つノード N があるもの全てについて, L の直前のノード N' から L への枝のタグが何であるかに応じて以下のいずれかを行う.

- タグが AX ならば, 補題 5.2 よりある w, t, t', t'' が存在して $N' = (w, t')$, $L = (w, t)$, $N = (w, t'')$ と書けるので, R_w から (t', t) を削除し (t', t'') を加え, St_w から state t を削除
- BEL ならば, 補題 5.2 よりある w, t, t', t'' が存在して $N' = (w', t)$, $L = (w, t)$, $N = (w'', t)$ と書けるので, B から (w', t, w) を削除し (w', t, w'') を加え, W から可能世界 w を削除
- DESIRE, INTEND であることは, 補題 5.2 からない

ここで, $t R_w t'$ を満たす t' の存在しないような全ての w と t に対し, (t, t) を R_w に加える. この場合, T のノード (w, t) の第 1・2 ラベルの $\rightarrow$ の左右には AX をトップレベルに持つ論理式はないことに注意. また, $(w, t, w') \in B$ を満たす w' が存在しないような全ての w と t に対し, (w, t, w) を B に加える. この場合も BEL について同様の注意が成り立つ. D, I についても同様の操作を行う.

最後に, 3.1 節の条件 (B-4) や (B-5) を満たさない w, w', w'' がある間, (w, t, w'') を B に加える (世界や

state は有限個なので有限の手間で済む). そのような (w, t, w'') を 1 つ $\mathcal{B}$ に加える前に, 性質

$(w, t, w') \in \mathcal{B}$ ならば, T のノード (w, t) の第 2 ラベルの $\rightarrow$ の左に現れる全ての $\text{BEL}(\phi)$ に対し, ノード (w', t) の第 1 ラベルの $\rightarrow$ の左には $\text{BEL}(\phi)$ と ϕ が現れる

が満たされているならば, 1 つ加えた後にもこの性質が満たされるので, この操作が全て済んだ後も, この性質は満たされている.

これにより M は $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}\text{-structure}$ になる.

6.2 完全性の証明

定理 6.1 体系 $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}$ は完全である.

証明 sequent S が証明可能でないとする. 6.1 節のアルゴリズムで作られた $(\text{B}^{KD45}\text{D}^{KD}\text{I}^{KD})_{CTL}\text{-structure}$ M は以下の性質を持つ.

$S_w \ni t$ とする. ‘手順 B’ で T のノード (w, t) を生成した段階での S''_i から S'''_{ij} に至る過程 (但し $w = w_0, t = t_0$ の場合は S から S' に至る過程) のどの sequent に対しても, その $\rightarrow$ の左に現れる論理式は (M, w, t) で真であり, 右に現れる論理式は (M, w, t) で真でない.

これを示すには, 論理式の構造に関する帰納法を, M の全ての state に対して同時並行で適用すればよい. 特に, $\text{A}(\phi \cup \psi)$, $\text{E}(\phi \cup \psi)$ に関しては次のように示せる.

- $\rightarrow$ の左の $\text{A}(\phi \cup \psi) \dots$ もしこれが偽なら, T の作り方から, (M, w, t) で始まるある path で, ずっと ϕ が真, ψ が偽となる. このとき, T のあるノード N とその子孫の葉ノード N' で, N から N' までの経路のノードの第 1 ラベルの $\rightarrow$ の左に常に $\text{A}(\phi \cup \psi)$ があり, しかもその経路の枝が全てタグ AX を持つものがある. すると, N の第 1 ラベルには, 終局性論理式 $\text{A}(\phi \cup \psi)$ を持つ導出のループを含む証明が存在することになり, 矛盾する.
- $\rightarrow$ の左の $\text{E}(\phi \cup \psi) \dots$ 同様.
- $\rightarrow$ の右の $\text{A}(\phi \cup \psi) \dots$ これに対しどこかで AU 右が適用されている. その次の sequent が AU 右

の前提の左だとすると, 帰納法の仮定から ϕ も ψ も (M, w, t) で偽なので $\text{A}(\phi \cup \psi)$ も偽. 一方, 右だとすると, 同じ理由で ψ は (M, w, t) で偽. また, $t R_w t'$ を満たすある (w, t') があって, それに対応するノードの生成中に, sequent の $\rightarrow$ の右に $\text{A}(\phi \cup \psi)$ が現れている. これに同様の議論を適用して, (M, w, t') で ϕ も ψ も偽か, あるいは ψ が偽でかつ $t' R_w t''$ を満たすある (w, t'') があって, 同じ議論が繰り返せる. すなわち, t で始まる w 上のある path があって, ずっと ψ が偽か, あるいはあるところで ϕ が偽でかつそこまでずっと ψ が偽である. 従って (M, w, t) で $\text{A}(\phi \cup \psi)$ は偽.

- $\rightarrow$ の右の $\text{E}(\phi \cup \psi) \dots$ 同様.

従って, 特に $(M, w_0, t_0) \not\models S$ である. よって, S は恒真でない. $\square$

7 判定アルゴリズム

6 章で示したアルゴリズムを, 証明可能な sequent に対しても適用すれば, sequent の証明可能性の判定アルゴリズムが得られる.

与えられた sequent S に Wang のアルゴリズムを準用し, $\neg/\vee/\text{AU}/\text{EU}$ 左・右を可能な限り適用する. その後, もし各 sequent の $\rightarrow$ の左同士か右同士に同じ論理式が複数回現れていれば, Weak でそれを 1 つに減らす.

この結果得られる sequent のうち, 様相オペレータを持たないか, S からの過程で既に現れている sequent と同じものについては何もせず, それ以外の各 sequent S' に対し, 6.1.1 節の ‘手順 B’ と同様に, 様相論理版の Wang のアルゴリズムを準用する. 得られた各 sequent S''_i に対し, S と同様に Wang のアルゴリズムを準用することを繰り返す.

こうして, 定義 4.4 の条件を満たす推論木が作れれば S は証明可能である. また, できなければ, 有限の手間でこの操作は行き詰まり, 証明可能でないとわかる.

現在, このアルゴリズムの Prolog によるインプリメントが動作している (<ftp://ftp.ics.nara-wu.ac.jp/pub/nide/research/bdi-1.00.tgz>).

8 今後の課題

複数のエージェントに関する BDI [5] をもつ logic への拡張も, エージェントの quantifier を入れなければ直ちに行えるだろう. また, mutual belief [5] のように, BDI オペレータの無限ネストに相当するオペレータを持つ体系に関しても, 本稿の AU/EU に対する取り扱いと同様の手法によって, 演繹体系を作れると考えられる. これらを導入できれば, マルチエージェントシステムの BDI logic を用いた仕様記述や検証に有効であろう. さらに, $\text{DESIRE}(\phi) \supset \text{BEL}(\text{DESIRE}(\phi))$ など, 合理的エージェントに要求される心的状態の整合性公理 [2] [3] の導入も検討を要する. それらは今後の課題である.

エージェントの quantifier に関する取り扱いや, また, 現時点ではまだ, ベースが命題論理の CTL に限定されているので, これを CTL* や述語論理に拡張できるかどうか今後の検討課題である.

参考文献

- [1] Emerson, E. A. and Srinivasan, J.: Branching Time Temporal Logic, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*(de Bakker, J., de Roever, W., and Rozenberg, G.(eds.)), Springer-Verlag, 1989, pp. 123–172.
- [2] Rao, A. S. and Georgeff, M. P.: Modeling Rational Agents within a BDI-Architecture, *Proc. of International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pp. 473–484.
- [3] Rao, A. S. and Georgeff, M. P.: Decision Procedures for BDI Logics, *Journal of Logic and Computation*, Vol. 8, No. 3(1998), pp. 292–343.
- [4] Singh, M. P., Rao, A. S., and Georgeff, M. P.: Formal Methods in DAI: Logic-Based Representation and Reasoning, *Multiagent Systems*, The MIT Press, 1999, pp. 331–376.
- [5] Wooldridge, M.: *Reasoning about Rational Agents*, The MIT Press, 2000.
- [6] 高田司郎, 五十嵐新女, 新出尚之, 榎本美香, 間瀬健二, 中津良平: マルチエージェント環境において意図的に言語行為を遂行する合理的エージェントの基本設計, *電気情報通信学会論文誌*, Vol. J84-D-I, No. 8(2001), pp. 1191–1201.
- [7] 萩谷昌己: *ソフトウェア科学のための論理学*, 岩波講座 ソフトウェア科学 11, 岩波書店, 1994.

マルチエージェントによるチーム航行支援

安藤 英幸 広野 康平 大和 裕幸

1 はじめに

ビジネスや製造のプロセスモデルを整理し、情報や業務の引継ぎの自動化を目指したワークフロー管理システムが注目されている。

ワークフロー管理システムの中心となるのは、プロセスモデルとそれを実行するワークフローエンジンである。一般にプロセスモデルは、コミュニケーション、作業の対象、ユーザーのアクティビティといった視点から有向ネットワークにより表現される。またワークフローエンジンは、プロセスのテンプレートからのインスタンスの作成、タスクの割り当て、実行とモニタリング、例外処理を行う。

これらの枠組みは、定型的にプロセスモデルを記述できる場合には十分である。しかし、とりまく環境の変化に対応してプロセスの更新が必要であったり、プロセスを外部環境との関連で表現するのが適している場合には、プロセスモデルの表現の工夫と、持続的にプランニング、スケジューリング、実行のモニタリングを行うワークフローエンジンが必要となってくる。

本研究では、船舶のブリッジ業務に動的なワークフロー管理システムを実装する事を目的としている。ブ

リッジの業務は、周囲の船舶や自船の位置といった状況に応じて、実行する業務を適応させる必要がある。このために必要となるプロセス表現の整理とワークフローエンジンの実装を行う。

2 背景

船上業務の種類は、例えば見張り、船位確認といったレベルの業務をユニットとして約 600~700 ユニット程度であると言われる [1]。それぞれの業務は、Hutchins も述べているように構造の良くわかった問題で、あらかじめ作業手順を定型化する事ができる [2]。しかし、それら定型的な業務も、自船位置と航路の関係、他船の位置と速度、ワッチレベル (watch level) と呼ばれる危険度の指標、他の作業、作業の記録といった多様なコンテキストに応じて、行うべき業務、優先順位、処理に要する時間、繰り返し業務に関する時間間隔等が変更する。

これらの船上業務についての詳細は SMS(Safety Management System) マニュアルという印刷物の形で各運航会社毎にまとめられ、船上で利用されている。しかし、作業の状況への依存性が高い事からマニュアルが複雑化したり、マニュアルの見直しに必要な実作業の履歴の入手が現状では困難であることから、ワークフロー管理システムに SMS マニュアルの情報を移植し、作業の見逃し、対応の遅れの防止、さらには、作業履歴の分析による基のプロセスモデルの改善が期待される。

Team Navigation Support with Multi-Agents System
Hideyuki Ando, 東京大学 新領域創成科学研究科, Graduate School of Frontier Sciences, University of Tokyo
Kouhei Hirono, (株) 日本海洋科学, Japan Marine Science Inc.

Hiroyuki Yamato, 東京大学 新領域創成科学研究科, Graduate School of Frontier Sciences, University of Tokyo

3 エージェントとアーキテクチャ

3.1 エージェントの構成

ブリッジ業務の動的なワークフロー管理システムは、プロセス、外的状況、作業に関する情報をネットワーク上で交換し、それらに基づいて適切な業務のプランニング、スケジューリング、実行モニタリングを持続的に行う必要がある。これらの実装にマルチエージェントを採用することとし、その実装のベースに、Jin らがワークフローによる協調設計支援のために開発した ActiveProcess 環境を利用する [3]。ActiveProcess の各エージェントは、ネームサーバーを介して KQML メッセージを交換する。コンテンツ言語としては、順序なしリストを表現できる独自のフォーマットを利用している。本研究では、航海士の支援を行うユーザーエージェント、船長につきチーム全体の業務プロセスを管理するマネージャーエージェント、船舶運航に関する情報をもつセンサーデータサーバーのラッパーとして働くリソースエージェントを配置した (図 1)。

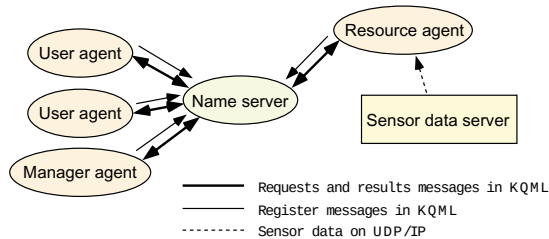


図 1 Multi agents configuration

3.2 多層型アーキテクチャ

ActiveProcess では、自律的な振る舞い、推論、エージェント間の協調といった知的な機構に関しては、ドメインに合わせて開発を行う必要がある。本システムでは、エージェントに求められるタスクの多様性から、即応、熟考、協調の 3 つの機能を独立したレイヤーとして実現し、協調ロボットの実装を良く整理した InterRAP アーキテクチャの概念を基本的に採用することとした [4]。ただし、熟考、協調層の推

論部分へのエキスパートシステム JESS(Java Expert System Shell) [5] の利用、各レイヤーのコンポーネントの構成の変更といった改良を加えた。

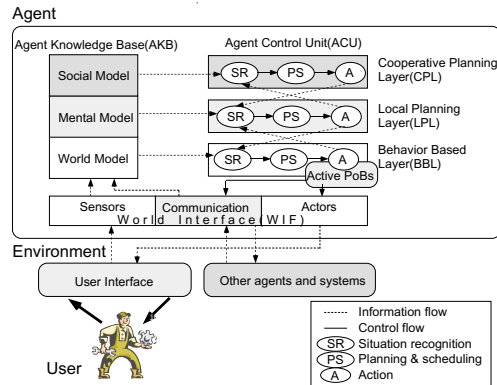


図 2 Agent architecture

エージェントの内部構成を図 2 に表す。全体は World Interface(WIF)、Agent Knowledge Base(AKB)、Agent Control Unit(ACU) によって構成される。WIF はエージェントと外部環境とのインターフェースで、ユーザーインターフェース、他のエージェントとの通信、センサーデータの取得に利用する。エージェントの持つ周囲の環境に関する知識、作業者の行うワークフロー、他エージェントに関する知識は World model, Mental model, Social model の 3 層に分けて AKB に置かれる。また、AKB はそれらの知識の登録、取り出しのためのインターフェースを持つ。ACU では AKB の知識を基に即応 (BBL)、熟考 (LPL)、協調 (CPL) のそれぞれの層が独立して状況の認識、プランニング、アクションを行う。ACU 内のレイヤー間の通信には KQML を使用する。即応層では、Pattern of Behavior(PoB) と呼ばれるエージェントの振る舞いのパターンを記述したモジュールの Active/Inactive の切り替えを行う。PoB には反射的な対応を行うための reactive PoB と他の層からの委託により Active な状態になる procedure PoB がある。

エージェントは全体として、AKB の知識をベースに即応、熟考、協調の各層がプランニング、スケジュー

リングを行い、PoB の実行リスト (Active PoBs) の切り替えを判断している事になる。

3.3 実装

システムは WindowsNT/2K 上で開発を行った。エージェントの開発には Java 言語を利用したが、他のアプリケーションとの連携を容易にするため Visual J++ を用いて COM(Component Object Model) の DLL(Dynamic Link Library) として実装した。また、ユーザーインターフェースの実装は Visual Basic を用いた。

4 プロセスマネージメント

4.1 プロセスモデル

Wilkins らは、現実問題へのプランニング技術の適用を考える場合、ドメイン知識の利用が不可欠である事を指摘している [6]。ドメイン知識を利用したプランニングの代表として HTN(Hierarchical Task Network) がある。HTN ではツリー構造によってプロセスを記述し、ゴールに合わせて上位概念を選択し、下位のツリーを展開し、最終的にプリミティブな操作で構成されるプランを得る。本研究でも、記述が可能であることから、階層構造というドメイン知識を用いたプロセス記述を採用し、特に Jin らが ActiveProcess で利用している Workflow, Activity, Work Element, Operation の 4 階層によるプロセスモデルに従う事とした。プロセスモデルのツリーの各階層の役割を、表 1 に示す。

また、動的な環境の変化に対応して適切なプランニングを行うために、全体のツリーのうちのサブツリーを駆動するかという知識をエキスパートシステム JESS のルールという形で記述する。(図 3)。

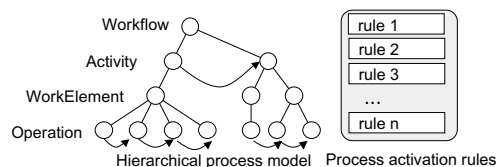


図 3 Process model

Workflow	チーム全体の作業プロセス
Activity	1 人の作業者に割り当てる作業の最小単位
Work Element	Operation の集合
Operation	作業の最小単位

表 1 Role of each layer

4.2 プランニング、スケジューリング

ユーザーの作業のプランニング、スケジューリングは、図 4 の手順で行われる。外部環境の変化にマッチするルールが発火し、実行すべきサブツリーが選択される。選択されたサブツリーは、順序の制約を満たすように展開され、一連のオペレーションにより構成されるプランとなる。これに、それぞれの作業時間を適応してスケジュールが構成される。この作業は継続的に繰り返される。また、現時点で、サブツリーの選択のために用意したルールの例を以下に示す。

- 周囲に船が接近してきた場合に起動
- ワッチレベルに応じた起動
- デフォルトで起動

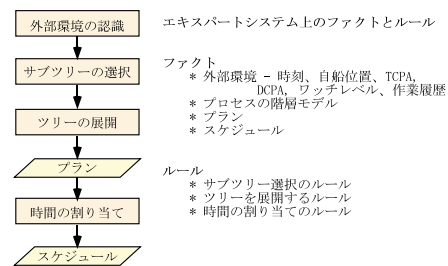


図 4 Planning and scheduling

4.3 実行とモニタリング

熟考層でスケジューリングが行われた作業は、ユーザーインターフェースを通して、ユーザーに対して提示され、その終了までをモニタリングされる。各作業のモニタリングは、図 5 に示す状態遷移規則に基づいて行われる。エージェントとユーザーインターフェー

スの間で、図 5 の矢印の上に示されたフラグを交換し、状態を遷移する。作業のモニタリングは、即応層のレベルの作業モニタリング用 PoB が担当する。開始時には熟考層から Active にするメッセージが即応層に送られ、ユーザーインターフェースを介してユーザーに提示される。作業終了時には、ユーザーからの終了の通知を受けて、熟考層が業務の終了を認識し、先のモニタリング用 PoB を Inactive にするように即応層にメッセージを送る。

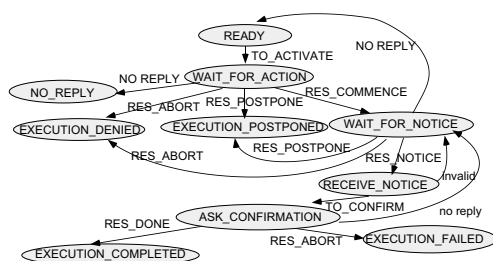


図 5 State transition for operation monitoring

5 検証

開発したエージェントの動作を確認するため、船舶のデスクトップ型の操船シミュレータ（仕様を表 2 に示す）に接続した。自船、他船に関する情報は、レーダー装置を代替するセンサーデータサーバからネットワーク上のリソースエージェントに対して送られる。シミュレーションの海域は、伊豆大島東方で、自船は洲崎沖変針点向け航行中とした。シミュレーションは、ワッチレベル 1（当直航海士 1 名体制）として開始した。当直航海士は船長から「見張り」、「船位測定」、「避航操船」の 3 つの Activity を与えられている。船長は船橋にいない状態であるが、「ワッチレベルの切り替え」の Activity が常時割り当てられている。この状態で、以下の点についてシステムの動作を確認した。

1. プロセスツリーの展開
2. 作業のモニタリング
3. 動的な環境でのサブツリーの選択

自船データ	1 秒間隔で生成
他船データ	3 秒間隔で生成（一般的なレーダーの周期が 3 秒であるため）。
自船の運動計算	舵角とエンジンテレグラフ値の入力より、応答遅れを考慮して船体運動計算する。
他船の運動計算	予め設定されたコースをトレースさせて運動を再現。変針動作は一定の角速度で回頭させている。
表示	電子海図上に自船位置と他船位置、それぞれの速度ベクトルを重ねて表示。ECDIS と同等

表 2 Specification of the simulator

5.1 プロセスツリーの展開

航海士エージェントは、「見張り」Activity 以下の階層構造を展開し、目視による監視、レーダーによる監視といった Operation を順次促し、次に「船位測定」Activity 以下の階層構造を展開し、下位の Operation の実行を提示し、当直航海士の作業遂行支援を行う事を確認した。

5.2 作業のモニタリング

航海士エージェントは、状態遷移規則に基づき、モニタリング用の PoB を通して、ユーザーインターフェースとの間でメッセージの送受信を行い、選択された Operation が最終的に、正常終了、異常終了、延期、応答なしのいずれかの状態になるまでモニタリングを行った。また、応答が許容時間内にない場合には、入力の催促を行った。

5.3 動的な環境でのサブツリーの選択

「ワッチレベル切り替え」Activity が起動される条件を、この検証では“ TCPA（最接近までの時間）がプラスで、DCPA（最接近時の距離）が 1 マイル以下となる船舶が、同時に、10 マイル以内に 2 隻以上現れたとき ”とした。船長エージェントは、リソースエージェントから TCPA, DCPA 情報を定期的に

取得し、この条件を満たす状態に変化した直後に、船長に対してワッチレベルの切り替え Operation を提示し、船長はワッチレベルを変更した。航海士エージェントはワッチレベルが変更された事を受け、「避航操船」Activity の担当を外し、代わって船長エージェントが「避航操船」Activity を担当するツリーに加えた。

6 まとめ

船上のブリッジ作業支援というドメインにおける動的ワークフロー管理システムの設計を行うにあたり、プロセスモデルとして、階層構造で表現されるプロセスモデルと、駆動するべきサブツリーを選択する規則を記述したエキスパートシステムのルールを併用した。また、それらを実行するためのワークフローエンジンのアーキテクチャとして、マルチエージェントを採用し、特に、多様なタスクを並行してこなすために、InterRAP の多層アーキテクチャを基に、エキスパートシステムを推論エンジンとして使用するなどの拡張を行った。

これらにより実装した動的ワークフロー管理システムが、環境を認識し、条件に適応するサブツリーを選択し、それを展開し、作業者のスケジュールを作成し、その実行をモニタリングする事を、船舶の操船シミュレーターを利用した実験において確認した。

実用に向けては、まだ非常に多くの課題がある。現時点では、プロセスモデルの準備とルールの数だけでなく、この整備が必須となる。また、エージェントの認識できる情報は大きく制限され、例えば気象や海象といった情報を扱うことができない。これらを含めて、どのように人間とエージェントが協調的に作業を行うかを考えていく必要がある。

ワークフローエンジンの機能としては、作業スケジュールの作成に関して、iterative repair の導入等による、より無理のないスケジュールの更新手法の検討、作業者間の負荷分散、ユーザーによるスケジュール変更を許すためのインターフェースが必要と考えている。

7 謝辞

本研究の遂行にあたり、(株)日本海洋科学の東京大学名誉教授小山健夫先生から多数のご助言を頂きました。また、東京大学助教授の白山晋先生、増田宏先生、海上安全技術研究所の伊藤博子氏には研究の遂行にご協力いただきました。システム開発にあたり、南カリフォルニア大学の Prof. Yan Jin からは ActiveProcess のソースコードの提供と多数のご助言を頂きました。ここに感謝いたします。なお、本研究は平成 12 年度文部省科学研究費補助金による基礎基盤 (A)(2)-1230817 の補助を受けました。

参考文献

- [1] 日本造船協会第 238 研究部会. 新しいフリートサポートシステムの研究、平成 10 年度報告書. 社団法人 日本造船研究協会, 1999.
- [2] E. Hutchins. The technology of team navigation, 1990.
- [3] Yan Jin, Li Zhao, and Arun Raghunath. Activeprocess: A process-driven and agent-based approach to supporting collaborative engineering. In *Proc. of ASME Design Engineering Technical Conference, Las Vegas, Nevada*, 1999.
- [4] Jörg P. Müller. *The design of intelligent agents: a layered approach*, Vol. 1177. Springer-Verlag Inc., New York, NY, USA, 1996.
- [5] Jess. <http://herzberg.ca.sandia.gov/jess/>.
- [6] D. Wilkins and M. desJardins. A call for knowledge-based planning. *AI Magazine*, Vol. 22, pp. 99–115, 2001.

Wizard of Oz法を使ったマルチキャラクタエージェントインターフェースの評価

辻本秀樹 北村泰彦 辰巳昭治

Character-agent interface is a user-friendly interface on the Internet. Most of the systems use a single agent alone, but we here consider the effect of multiple character-agents. We have developed an information retrieval system called Venus & Mars based on multiple character-agents interface. In this paper, we observed and compared interactions between users and agents in two versions of Venus & Mars; one version employs a single character only and another employs multiple characters, by using Wizard of Oz method.

1 はじめに

今日ではパソコンやインターネットが社会に浸透し、今までコンピュータに触れることの少なかった初心者ユーザが利用する機会も増えている。そのような初心者ユーザを含め、ユーザに優しいインターフェースとしてキャラクタエージェントインターフェースがある。またユーザ層が広がることで情報源も広がり、分散した情報の統合が有効となる。ここで、キャラクタエージェントインターフェースのほとんどがシングルエージェントによるものであることから、それらのシステムを組み合わせでマルチキャラクタエージェントインターフェースのシステムとして動作させるということが考えられる。このマルチキャラクタエージェントインターフェースの概

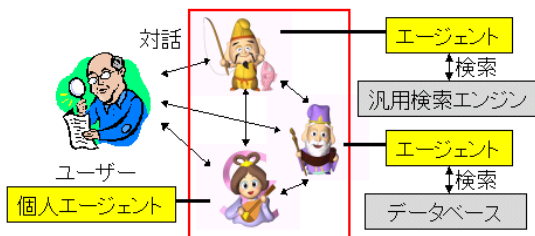


図1 マルチキャラクタエージェントインターフェース

観を図1に示す。

このようなマルチキャラクタエージェントインターフェースを実装し、それぞれのエージェントが持っている情報を組み合わせて検索を行うことができる、Venus & Mars[1]という情報検索システムがある。本研究ではマルチキャラクタエージェントインターフェースが分散した情報を統合するシステムに有効であるかを調べるため、Venus & Marsのインターフェースをシングルエージェントとマルチエージェントで動作させ、それぞれのユーザとエージェントのインタラクションをWizard of Oz法[3][4][5]を用いて観測し、マルチキャラクタエージェントインターフェースを評価した。これはシステムのインターフェース設計に生かせると考えられる。

2 Venus & Mars：協調型マルチエージェント検索システム

Venus & Marsは情報検索に不慣れなユーザでも、自然言語でよりの確なレシピの検索を可能とすることを目標としたもので、Microsoft Agentを利用したマルチキャラクタエージェントインターフェースが実装されて

Hideki Tsujimoto, Yasuhiko Kitamura, Shoji Tsumi, 大阪市立大学大学院 工学研究科 情報工学専攻, Department of Information and Communication Engineering, Graduate School of Engineering, Osaka City University.



図2 Venus & Mars の動作画面

いる。エージェントには、レシピまたは健康に関する知識を持ちその領域での情報検索や助言を行う専門エージェント（献立検索エージェント、健康物知りエージェント）と、利用者のプロフィールを情報検索に反映させる個人エージェントが存在する。その動作画面を図2に示す。

ユーザは話しかけたいエージェントをクリックし、自然言語のテキストで発話することができる。ユーザや他のエージェントの発話を取得したエージェントはそれぞれ独立に処理を行い、音声による発話と身ぶりを交えて応答する。このときユーザやエージェントの発話は全てのエージェントに伝わっている。

Venus & Mars で登場するエージェントの役割とその行動は以下のようになっている。

- 献立検索エージェント「コン」：レシピを検索するエージェントで、食材、料理名、味付けに関するキーワードを知識として持っている。ユーザまたは他のエージェントの発話に自分の知識と一致するキーワードが有れば、そのキーワードと検索隠し味 [2] と呼ばれる追加キーワードを用いて検索する。また、検索結果が多い場合には絞り込み検索を行うためのキーワードを尋ねる。
- 健康物知りエージェント「チョー」：健康に関する情報を提供するエージェントで、食材、病気や症状に関するキーワードと、食材が持つ効能、病気や症状に対して効果のある食材を知識として持つ。ユーザまたは他のエージェントの発話に自分の知識と一

致するキーワードが有ればそのキーワードについて発話する。

- 個人エージェント「ペッコ」：ユーザの入力を監視するエージェントで、絞り込み検索でユーザが入力した食材や味付けに関するキーワードをユーザの好みとして記憶しておく。献立検索エージェントが絞り込み検索を行うためのキーワードを尋ねたときにそのキーワードを発話する。

これらのエージェントは検索時に2つの点で協調する。

1つはコンが絞り込み検索を行うときにペッコがユーザの好みを発話することで、自動的にユーザの好みによる絞り込み検索を行えるように支援する点、もう1つは健康に関するキーワードが入力されときにチョーがそれに対して食材を発話することで、健康に関する知識の無いコンがレシピ検索を行えるようになる点である。

3 評価

本研究ではマルチキャラクターエージェントインターフェースの評価のため、Venus & Mars において、そのインターフェースがシングルエージェントとマルチエージェントの場合でのユーザとエージェントのインタラクションを観測し比較した。

3.1 Wizard of Oz

当初行った Venus & Mars をそのまま用いた実験では、それぞれのエージェントの持つ知識が少なく、入力に対する応答や協調が上手くできない場合が多かったため、インタラクションのほとんどがレシピ検索のキーワード入力という結果となった。このような状態を避けるためには、エージェントはより多くの知識を持ち、自然言語での入力を正しく解釈して応答する必要がある。しかし、現在の技術ではそのような機能を実装するのは難しいため、Wizard of Oz 法を用いて評価を行うことにした。

Wizard of Oz 法では wizard と呼ばれる人間のオペレータがシステムの一部分または全体をシミュレーションし、見かけ上はコンピュータによるシステムとして動作させ、被験者の行動を観測する。ここではオペレータがエージェントを操作するように Venus & Mars を改造した。オペレータには被験者とオペレータの発話内容

が見えており、それが返事できる内容のものであれば応答し、そうでなければ理解できなかったことを被験者に分からせるように発話する。オペレータが返事できる内容は、レシピ、健康、エージェントの個人的なこと、被験者の好み、挨拶などの一般的な事柄とした。

3.2 評価方法

評価は次の2種類の実験により得られた結果を比較することで行った。

- マルチエージェント

前述のエージェント3体が登場する。最初に各エージェントは自分の役割を明らかにするためにそれぞれ自己紹介を行う。特にコンとチョーは、「献立探しの名人」、「食は健康の基。いつでも健康でいられるように、体にいいものを教えてしんぜようぞ。」という台詞で役割を示している。ここでは3人のオペレータがそれぞれの担当するエージェントを2章で述べたエージェントの役割と行動に従って操作する。

- シングルエージェント

前述のエージェントの能力を全て持っているエージェント1体（見た目はコンのものである）が登場する。献立検索と健康についての知識があることを示すため、このエージェントの自己紹介では上記の台詞を両方とも用いている。また、この場合もマルチエージェントの場合と同様に3人のオペレータがエージェントを操作するが、レシピについての話題はコンのオペレータ、健康についての話題はチョーのオペレータ、それ以外の話題はペッコのオペレータが操作するようにした。

被験者には自然言語で会話のできるキャラクタを利用したレシピ検索システムであると説明し、どちらか片方のタイプを割り当てて実際に使ってもらう。被験者からエージェントへの発話回数が20回を越えるのを目安として1回の実験とした。また、オペレータがエージェントを操作していることは実験終了後に伝えている。

3.3 実験結果と考察

3.2節で記した方法で、大阪市立大学の学生10人に実験を行った。実験では被験者のパソコン歴、インターネット歴、レシピへの関心、健康への関心をアンケート

調査し、その傾向や性別が偏らないように振り分けた。マルチエージェントの被験者の1人を除き、全員20回以上の発話を行った。各被験者ののはじめの20回の発話を整理した結果を表1～3に示す。

表1 被験者の発話の種類

	マルチ	シングル
質問	34(36.2%)	49(49.0%)
検索依頼	43(45.7%)	29(29.0%)
その他	17(18.1%)	22(22.0%)
合計	94(100.0%)	100(100.0%)

表2 被験者の発話内容

	マルチ	シングル
エージェント	24(25.5%)	32(32.0%)
レシピ	40(42.6%)	33(33.0%)
健康	13(13.8%)	3(3.0%)
好み	3(3.2%)	3(3.0%)
その他	14(14.9%)	29(29.0%)
合計	94(100.0%)	100(100.0%)

表1は被験者の発話を種類によって「質問」、「検索依頼」、「その他」に分類し、表2は内容によって「エージェントについての個人的な話題」、「レシピについての話題」、「健康についての話題」、「被験者の好みについての話題」、「その他」に分類したものである。表の数値はそれぞれの分類について、被験者の発話回数とその割合を示している。表3はマルチエージェントの実験において、被験者が表2で分類した内容をどのエージェントを対象として発話したかを示している。表の数値は発話内容とその対象となるエージェントについて、被験者の発話回数とその割合を示している。

表1からマルチエージェントとシングルエージェントのどちらにおいても検索依頼以外の発話が多く発生しており、そのほとんどがエージェントについての質問であったことから、エージェントは被験者の気をひく存在であったと考えられる。また表2から、マルチエージェントではレシピについてや健康についての検索に関係す

表 3 マルチエージェントでの被験者の発話内容とその対象

	エージェント	レシピ	健康	好み	その他
ペッコ	8(33.3%)	9(22.5%)	3(23.1%)	0(0.0%)	11(78.6%)
コン	7(29.2%)	29(72.5%)	2(15.4%)	3(100.0%)	3(21.4%)
チョー	9(37.5%)	2(5.0%)	8(61.5%)	0(0.0%)	0(0.0%)
合計	24(100.0%)	40(100.0%)	13(100.0%)	3(100.0%)	14(100.0%)

る発話が多いのに対し、シングルエージェントではエージェントについてやその他の検索に関係しない発話が多い。このことよりシングルエージェントでは被験者を飽きさせやすかったと考えられ、マルチエージェントの方が検索に関係した発話を発生させやすいと言える。

また表 3 から、マルチエージェントにおいて被験者は、レシピについては献立検索エージェントに、健康については健康物知りエージェントに、その他の話題は個人エージェントに発話することが多く、エージェントを使い分けていたと考えられる。これらのことから、マルチエージェントにおいて被験者は各エージェントの役割を認識し、それによってマルチエージェントの方が検索に関係した話題を発話が多く発生したと考えられる。また、どちらの場合においても被験者の好みについての話題は少なかったが、マルチエージェントでは個人エージェントにその他の話題を発話することが多かった。これは個人エージェントは司会者のような役割として認識され、ユーザの好みを覚えておくという役割についてはあまり重要視されなかったか、上手く認識されなかったと考えられる。この点は、個人エージェントが被験者の好みを覚えていることをよりアピールすれば変化が見られるかもしれない。

4 まとめ

本研究では、情報検索システムにマルチキャラクタエージェントインターフェースを実装した Venus & Mars において、シングルエージェントとマルチエージェントの場合のユーザとエージェントのインタラクションを Wizard of Oz 法を用いて観測し比較した。その結果、マルチエージェントの方が検索に関係する内容の発話が発生しやすく、ユーザを飽きさせない効果があると考えられた。さらに、マルチエージェントにおいて被験者は各エージェントの役割を認識しており、エージェントを使

い分けていることが分かった。これらの結果は、Venus & Mars のような分散した情報を統合して提供するシステムにとってマルチキャラクタエージェントインターフェースが有効であることを表している。しかし、エージェントの役割が意図したものと違って認識される可能性があることも示していた。このことから、エージェントを設計する際にはその役割をはっきりさせる必要があると言える。

現在は実験によって得られたシングルエージェントとマルチエージェントの場合での被験者のインタラクションの違いがどのような理由から起きたのか、また被験者がエージェントの役割を認識するまでにどのようなインタラクションを行っているかという視点から結果の解析を続けているところである。

参考文献

- [1] Yasuhiko Kitamura, et al.: Interactive Integration of Information Agents of the Web, Matthias Klusch, Franco Zambonelli (Eds.), Cooperative Information Agents V, Lecture Notes in Artificial Intelligence 2182, Berlin et al.: Springer-Verlag, pp.1-13,(2001).
- [2] Satoshi Oyama, et al: Keyword Spices: A New Method for Building Domain-Specific Web Search Engines, Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), pp.1457-1463,(2001).
- [3] Yeonsoo Yang, Masayuki Okamoto, and Toru Ishida: Applying Wizard of Oz Method to Learning Interface Agent, Proceeding of Workshop on Software Agent and its Applications(SAA2000), pp.223-230,(2000).
- [4] Malsby, D., Greenberg, S. and Mander, R., Prototyping an Intelligent Agent through Wizard of Oz, Proc. of the conference on Human factors in computing systems, pp.277-284,(1993)
- [5] Dahlback, N., Jonsson, A. and Ahrenberg, L., Wizard of Oz studies - Why and How, Proc. of the international workshop on Intelligent user interfaces, pp.193-200,(1993)

エージェント基盤のための通信プロトコル： Agent Platform Protocol

高橋 健一 鍾 国強 雨宮 聡史 峯 恒憲 雨宮 真人

近年、ネットワークで接続された複数の機器が協調や交渉を行い仕事を達成するシステムを作る枠組みとして様々なエージェントが提案されている。これらのエージェントシステムでは、エージェントの活動を促進する環境（AP）が個々に準備されているが、それら異なる AP 上のエージェントシステムを相互運用する手段については提供されていなかった。そこで、本稿では、AP 間のプロトコルとして Agent Platform Protocol (APP) を提案する。APP を用いることで、1) サーバを介さない P2P な AP 間通信と 2) 物理的環境を意識する必要のないエージェントコミュニケーションを実現できる。

1 はじめに

インターネットの普及とともに、ネットワークで接続された複数の機器によって実現された多くの情報システムやサービスが提供されてきた。これらのシステムでは、分散された機器が互いの情報を交換することで協調や交渉を行い問題を解決する必要が出てきており、このようなシステムを作る枠組みとして様々なエージェントが提案されている。

エージェントシステムでは、エージェントの活動を促進するための環境（Agent Platform, 以降 AP と記す）（例えば、KODAMA の ACZ [1] [2] や FRM の Field [3], Hive の cell [4] など）が準備されている。エージェントは、AP 上で、仕事の達成に必要な情報を得るためのコミュニケーションや仕事の達成に必要な資源を得るための AP 間の移動を行う。これらの AP は、元々各エージェントシステム固有のものとして開発されているため、そのような異なる AP を持つエージェントシステムを相互運用することは難しい。エージェントの標準化促進団体 FIPA [5] では、Agent Communication Language (以降、FIPA ACL) や、Agent Platform などの仕様が定められているが、その FIPA ACL では、メッセージの送信側と受信側エージェントの物理的なアドレスを指定する、受信側エージェントの物理的位置を明確に意識した仕様となっている。しかし、これは、次のような弊害を引き起こす。

- モバイルエージェントとのコミュニケーションが困難。

エージェントが移動すれば、エージェントが存在するアドレスも変わる。すなわち、受信側エージェントの物理アドレスを特定することは難しい。

- エージェントに、予め決定された連携を行わせることが難しい。

例えば、2 種類のエージェント、Ping と Pong が存在し、Ping は、適当なタイミングで Pong にメッセージを送り、Pong の存在を確認する仕事を持つとする。このとき、2 種類のエージェント

Communication Protocol for Agent Platform : Agent Platform Protocol.

Kenichi TAKAHASHI, Zhong GUOQIANG, Satoshi AMAMIYA, 九州大学 大学院 システム情報科学府, Graduate School of Information Science and Electrical Kyushu University.

Tsunenori MINE, Makoto AMAMIYA, 九州大学 大学院 システム情報科学府, Graduate School of Information Science and Electrical Engineering, Kyushu University.

が別々にインストールされ、予めそれぞれのアドレスを特定できない場合、Ping は、Pong の物理的なアドレスを知ることができないため、送信先としては、"Pong"とだけしか指定できない。しかし、これでは、Pong が存在するアドレスが指定されていないため、Ping から Pong にメッセージを伝達するには、Pong の物理的位置を検索する機構が必要となる。

これらの問題を解決するためには、まず AP 間に共通のインタフェースを提供するためのプロトコルを定義しなければならない。そして、そのプロトコルでは、次のことを想定しなければならない。

- AP にエージェントの物理的地址を管理させ、エージェントに物理的な情報を意識させないようにする。
- 複数の種類の AP が存在しても、各 AP が一意にエージェントを識別できなければならない。
- AP によって、エージェントの管理方式が異なる。
- 要求した機能が、要求先 AP では実現されていない可能性がある。
- 一般に公開されていなければならない。

そこで我々は、AP 間でエージェントのコミュニケーションをサポートするためのプロトコルとして Agent Platform Protocol (以下、APP) を提案する。APP では、13 の method と header が定義されたリクエスト／レスポンス指向のプロトコルであり、AP はそれらに従ったメッセージを用いて他の AP と協調を行う。APP は、次の機能を提供する。

- サーバを介さない Peer to Peer (以降、P2P) な AP 間ネットワーク。
- エージェントメッセージの送受信機能。
- エージェントメッセージの伝達先の検索機能。
- エージェントの移動。

AP は、APP に従ってメッセージの送信先エージェントを検索し、メッセージの伝達を行うため、エージェントは、互いの物理的なアドレスを意識せず、コミュニケーションを行うことができる。エージェント開発者は、AP に APP を実装することで、他で開発された異種の AP との通信が行えるようになり、様々な場所で開発された様々な種類のエージェントを、互

いに利用できるようになる。

以下、2 節では APP について述べ、3 節で APP を適用した AP の例として ACZ を紹介する。そして、4 節で APP を適用した ACZ を用いて、APP の機能についての検証実験を行った結果について報告する。

2 Agent Platform Protocol (APP)

APP は、エージェントの活動をサポートする AP 間の通信プロトコルであり、図 1 のように位置づけられる。APP は、低レベルの通信プロトコルとして TCP [8] を持つ。TCP では、信頼性のあるコネクション指向型の通信を提供しており、データの欠損や重複、届けられる順序の違いなどを発生しない。

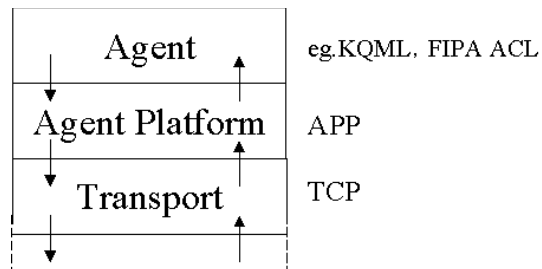


図 1 APP の位置づけ

APP の主な特徴として次の点が上げられる。

- サーバを介さない P2P な AP 間ネットワークの構築。

P2P なネットワークでは、各ネットワークの構成要素が置き換え可能なため、耐故障性が高く、一つの AP の故障が AP のネットワーク（以降、AP で構成されるネットワークを AP ネットワークと呼ぶ）に与える影響は少ない。また、サーバ／クライアントモデルで構成された AP ネットワークのように、サーバの役割を担う AP に負荷が集中するという問題はなく、更に、サーバの役割を担うような特定の AP への接続は不要で、AP ネットワークの任意の一つの AP への接続で AP ネットワークに加わることができる。

- エージェントメッセージの送受信機能の提供。

エージェントはコミュニケーション手段として、メッセージ（以降、APP によって用いられるメッセージ

表 1 APP で準備された method 一覧

分類	method	説明
Connect	LINK	AP 間の認証及び通信経路の作成
	DeLINK	AP 間の通信経路切断
	GET_LINK	要求先の AP が持つ通信経路情報の要求
	ALIVE	要求先が活動状態であるか否かの確認
Deliver	DELIVER	エージェントメッセージの伝達
	LOCATE_AGENT	エージェントの検索依頼
	GET_AGENT_LIST	要求先の AP が持つエージェント情報の要求
	PUT_AGENT_LIST	保持しているエージェントの情報を他の AP に通知
Mobile	GET_AGENT	エージェントを要求先の AP からローカルな AP に移動させる
	PUT_AGENT	ローカル AP 上のエージェントを要求先の AP に移動させる
Reply	ACCEPT	返答. 要求を受理したときに用いる
	SUCCESS	返答. 要求が成功したときに用いる
	FAILURE	返答. 要求が失敗したときに用いる

と区別するために、エージェントメッセージと呼ぶ)を用いた情報交換を行う。

- エージェントメッセージの伝達先の検索機能の提供。

AP は、エージェントメッセージの送信先エージェントのアドレスが分からない場合、エージェントメッセージの伝達先を探さなければならない。エージェントメッセージの伝達先検索のため、APP では、(1)他の AP に検索を任せる方法と(2)他の AP から情報を集め自分自身で検索を進める方法とを提供する。

- モバイルエージェントの移動をサポート。

エージェントに、仕事の達成に必要な資源を求め AP 間を移動する手段を提供する。

2.1 AP メッセージ

APP で用いられるメッセージを AP メッセージと呼ぶ。AP メッセージは、method, header, body から構成され、この順に記述される。ここで、method は、AP メッセージが表す要求、または、応答を表す。header は、method を解釈、処理するのに必要な情報である。body は、各 method が対象とする任意のデータである。

AP メッセージ = {method, header, body}

2.1.1 method

AP は、エージェントのコミュニケーションや移動をサポートする環境を提供する。このため、APP では、13 の method を準備している(表 1)。

Connect, Deliver, Mobile に分類される method は、AP に対して要求を出し、その応答として、AP は Reply に分類される method を用いる。

Connect サーバを介さないで P2P な AP ネットワークを構成するための method 群。LINK によって、AP 間での認証を行い、お互いの存在を知り、相互に信頼性のある通信を可能とする。GET_LINK は、要求先の AP が接続している AP の情報を要求する。AP は起動時に、まず LINK method を使って任意の AP に接続し、AP ネットワークに加わる。次に GET_LINK で AP ネットワークの情報を求める。新たに得た AP ネットワークの情報を利用して、複数の LINK を行うことで、特定の AP への依存度を下げることができる。DeLINK は、要求元の AP が通信を保証できなくなることを伝える。DeLINK を受け取った AP は、接続可能な他の AP との接続が分断されないように GET_LINK や LINK method を使って、ネットワークが分断されないように対処した後、DeLINK への応答を行わな

ければならない. ALIVE は, AP が実際に AP メッセージを受け付けることが可能であるかの確認を行う.

Deliver エージェントメッセージを送受信するための method 群. DELIVER は, エージェントメッセージを伝達する method では, body 部にエージェントメッセージを指定し, aimAt header(表 2 参照) でエージェントメッセージの受信側エージェントを指定する. DELIVER を受信した AP は, aimAt header が示すエージェントに body 部のエージェントメッセージを伝達する. 受信側エージェントが存在する AP が分からない場合の検索手段として, LOCATE_AGENT と GET_AGENT_LIST を準備している. LOCATE_AGENT は, 他の AP に検索を任せる方法であり (図 2(a)), LOCATE_AGENT を受信した AP は, aimAt で指定されたエージェントが存在する AP を探し出し, 見つければ, body 部をそのアドレス (IP アドレス/Port 番号) とした SUCCESS を返す. GET_AGENT_LIST は, 要求先の AP に, それを持つエージェントの情報を要求する. これは, 様々な AP からエージェントの情報を集め, その情報を用いて目的のエージェントが存在する AP を自分で見つける場合に利用する (図 2(b)). また, PUT_AGENT_LIST

は, 保持しているエージェントの情報を他の AP に通知する場合に利用する.

Mobile エージェントの移動を行うための method 群. POST_AGENT は, 要求先の AP にエージェントを移動させ, GET_AGENT は, 要求先の AP からエージェントを移動させる.

Reply Connect, Deliver, Mobile の各 method の応答として用いられる. ACCEPT は, 各 method を受理したことを表し, FAILURE は, 要求が失敗したことを, SUCCESS は, 要求が成功したことを表す.

2.1.2 header

APP における header は, その役割によって, general-header, body-header, authentication-header に分けられる (表 2). general-header は, すべての AP メッセージで必須であり, body-header は, body 部を持つ AP メッセージに必要とされる. authentication-header は, AP 間の認証が必要なきに用いられる. header は,

header = header-name ":" value

の形で表される.

2.1.3 body

body は, method で必要とされる任意のデータである. 例えば, DELIVER の場合, body は, エージェントメッセージとなる.

3 APP の適用例

我々の研究室で研究・開発しているマルチエージェントシステム KODAMA (Kyushu university Open & Distributed Autonomous Multi-Agent) では, AP として ACZ (Agent Communication Zone) を利用する. 本節では, まず, ACZ について述べ, 次に ACZ への APP 適用例を示す.

3.1 ACZ

ACZ は, 論理的なエージェント空間を物理的な構造にマッピングする機構であり, エージェント間のコミュニケーションを実現するために必要となる物理的通信機能を提供する. 通常, 一つの計算機上に一つの

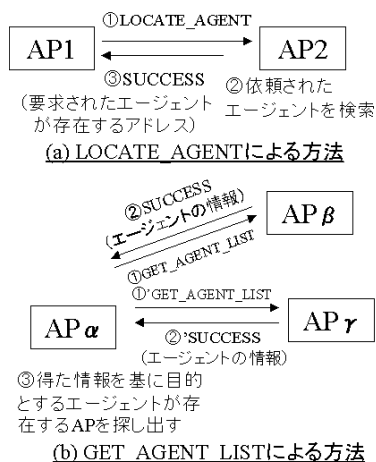


図 2 エージェントの検索方法

表 2 header 一覧

	header-name	value	説明
general-header	message-id	string	メッセージつけられた固有の ID
	from	AP-Address (IP-Address/port)	メッセージを送信した AP のアドレス
	via	*AP-Address	メッセージの経路
	refer	*message-id	メッセージの動機となった message-id
	AP-Type	string	AP のタイプ
authentication-header	authentication-type	string	認証方法
	authentication-value	string	認証キー
	authentication-expire	number	authentication-value の有効期限
	absolute	0 or 1	リンクの重要性
body-header	body-size	number	body 部のサイズ
	body-type	string	body 部のデータ形式
	aimAt	string	body 部のデータの対象

* は、0 回以上の繰り返しを表す。

ACZ が存在し、分散環境上では複数の ACZ が存在する。これらの ACZ は、ネットワークを構成し、分散した計算機の間での通信を制御する。この管理のために ACZ は、ローカル層とリモート層より構成される (図 3)。

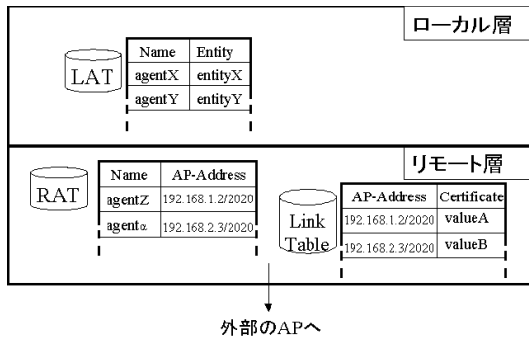


図 3 ACZ の構造

ローカル層は、ローカル ACZ 上のエージェントを管理するテーブル LAT (Local Agent address Table) を持ち、ローカル ACZ 上のエージェントへのエージェントメッセージの伝達を行う。ローカル ACZ 上のエージェントの生成または消滅時には、その名前が LAT に追加／削除される。

リモート層は、他の ACZ 上で活動するエージェントのアドレスを記したテーブル RAT (Remote Agent address Table) と、他の ACZ の認可情報を記したテーブル LT (Link Table) を持ち、他の ACZ 上で活動するエージェントへエージェントメッセージを伝達する際に利用される。

3.2 ACZ への APP の適用

ACZ のライフサイクルは、図 4 のように示される。

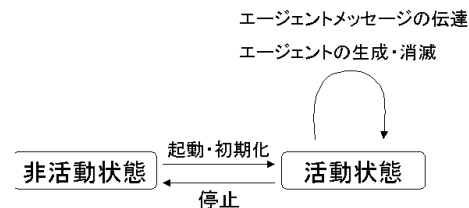


図 4 ACZ のライフサイクル

ACZ は生成されると直ぐに起動・初期化され、非活動状態から活動状態になる。活動状態中は、エージェントの生成や消滅、エージェントメッセージの伝達が行われる。そして、最後に停止され、活動状態から非活動状態に変化する。このように、ACZ の動作は、

起動・初期化、エージェントの生成・消滅、エージェントメッセージの伝達、停止の4つに分けられる。このうち、エージェントの生成・消滅はローカル層での動作であり、APP は関係しない。そこで、エージェントの生成・消滅以外の3つの ACZ の動作について APP がどのように適用されているかを示す。

3.2.1 ACZ の起動・初期化

KODAMA は、Java で開発されており、ACZ は次のコマンドで起動・初期化される。

```
java acz.ACZ (使用 port) (接続 IP:port)
```

図 5 のような状態において、

```
java acz.ACZ 2000 192.168.0.1:2020
```

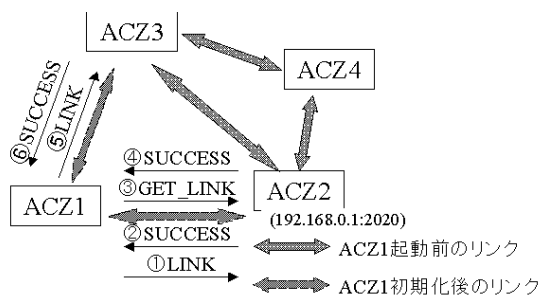


図 5 ACZ の初期動作

で ACZ1 を起動すれば、ACZ1 は、まず、LINK を使って、ACZ2 (192.168.0.1 / 2020) との間で、相互通信のためのリンクを作る。これは、ACZ1 と ACZ2 が各自の LT に相互登録する (図 5 ①, ②) ことによって実現する。次に、ACZ2 の持つ ACZ ネットワークの情報を集め (図 5 ③, ④)、集めた ACZ 中の任意の ACZ との間で相互の通信経路を作成する (図 5 ⑤, ⑥)。図 5 では、ACZ3 とだけ通信経路を作成しているが、ACZ の設定^{†1}によって、任意の数の通信経路を作成することができる。このことで、ACZ1 は特定の ACZ への依存度を減らすことができる。

3.2.2 ACZ の停止

図 6 のような状態において、ACZ1 が停止すれば、ACZ2 と ACZ3 の間での通信経路が無くなり、ネットワークが 2 つに分断される。このため、ACZ1 か

ら DeLINK を受け取った ACZ2 は (図 6 ①), ACZ ネットワークの分断をさけるため、GET_LINK で停止する ACZ1 から ACZ ネットワークの情報を受け取り (図 6 ②, ③), ACZ3 との間にリンクを構成する (図 6 ⑤)。

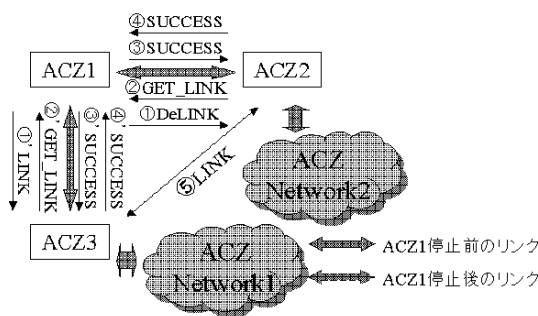


図 6 ACZ の停止動作

3.2.3 メッセージの伝達

エージェントメッセージの受信側エージェントが LAT のエージェントであれば、ローカル層でエージェントメッセージの伝達を行う。RAT に登録されているエージェントであれば、RAT からアドレスを求め、DELIVER を用い、エージェントメッセージを伝達する。LAT, RAT 双方に登録されていないエージェントであれば、そのエージェントの検索が行われる。

ACZ でのエージェントの検索は、KODAMA エージェントの論理的構造を参照し、LOCATE_AGENT を用いて行う。(図 7)。KODAMA では、エージェントは階層的に配置され、エージェントの名前が階層の場所を示す。例えば、"root・A" の名前を持つエージェントは、"root" の名前を持つエージェントを上位階層に持ち、"root・A・child" の名前を持つエージェントを下位階層に持つ。ACZ は、検索依頼を表す LOCATE_AGENT を用いて、次の方針でエージェントの検索を行う。

- (a) 検索は RAT, LAT から refer header に含まれない ACZ で活動するエージェントで階層的に最も近いエージェントを見つけ、見つけたエージェントが目的のエージェントであれば、始めに LOCATE_AGENT を出した ACZ に SUC-

^{†1} ユーザによって、ACZ 初期化時に作成される通信経路の枠組みが指定される。

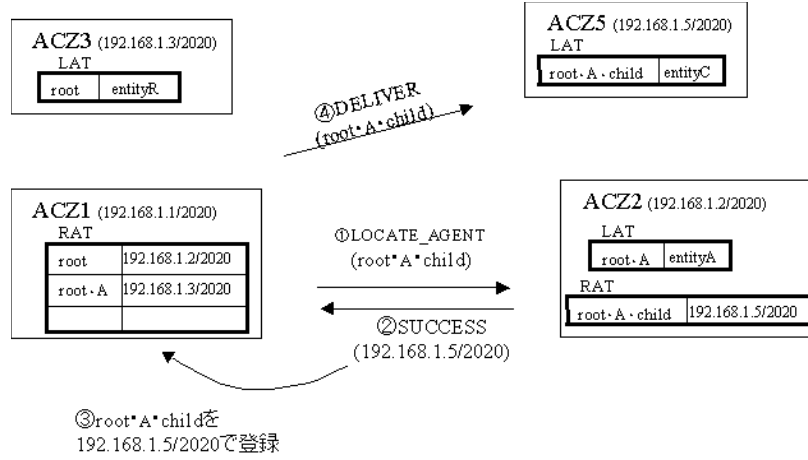


図 7 ACZ でのエージェント検索方法

CESS で応答し (図 7 ②), 目的のエージェントでなければ, 見つけたエージェントが活動する ACZ に LOCATE_AGENT で検索を依頼する (図 7 ①). LAT, RAT が共に要素を持たない場合は, 最初に LT に登録された ACZ に LOCATE_AGENT で検索依頼を出す.

(b) LOCATE_AGENT を受け取った AP は, via header を見て, すでに受け取っている LOCATE_AGENT 要求であれば, 始めに LOCATE_AGENT を出した ACZ に FAILURE で応答する. まだ, 受け取っていない LOCATE_AGENT 要求であれば, (a) を行う.

(c) LOCATE_AGENT の応答として SUCCESS を受け取った ACZ は (図 7 ③), 検索結果を RAT に登録する. 応答として FAILURE を受け取れば, 検索失敗である.

結果として, ACZ は, 目的のエージェントを見つけた場合, DELIVER で (図 7 ④), エージェントメッセージを伝達する.

4 実験

APP の有効性を検証するために, APP を実装した ACZ で実験を行った. 実験では, 各計算機に一つの ACZ を準備し, 合計 8 台の計算機を利用した. 各 ACZ は, ランダムに停止・起動を繰り返すこととし, 起動時には接続する ACZ を活動中の ACZ からラン

ダムに選び, 停止時には LAT, RAT, LT のテーブルを初期化した. また, 各 ACZ には, 10 のエージェントを準備し, それぞれに適当なエージェントとコミュニケーションを行わせることとした. この条件で 1 時間経過したあとにすべての ACZ を停止させた. このときの実験結果を表 3 に示す.

表 3 実験結果

総 AP メッセージ数	2831
総起動・停止回数	44
エージェントのコミュニケーション数	906

4.1 ネットワークに関する評価

実験では, ACZ の起動・停止が共に 44 回行われた. IP=192.168.35.76, port=2200 の ACZ 停止時に発生した AP メッセージを例 1 に示す.^{†2}

例 1 では, 192.168.35.76 の停止によって AP ネットワークが分断されることを防ぐため, GET_LINK で AP ネットワークの情報を求め, 192.168.35.64 との間に新しくリンクを作成した. 192.168.35.73 との間には, すでにリンクを持っていたため, 新しくリンクは構成しなかった.

結果, 44 回の ACZ の起動・停止にも係わらず, AP

^{†2} すべての ACZ で 2200 番ポートを使用しているため, ポート番号は省略した.

1.DeLINK:From 192.168.35.76 to 192.168.35.65
 2.ACCEPT:From 192.168.35.65 to 192.168.35.76 --(1の応答)
 3.GET_LINK:From 192.168.35.65 to 192.168.35.76
 4.SUCCESS:From 192.168.35.76 to 192.168.35.65,
 [192.168.35.73, 192.168.35.64] --(3の応答)
 5.SUCCESS:From 192.168.35.65 to 192.168.35.76 --(1の応答)
 6.LINK:From 192.168.35.65 to 192.168.35.64
 7.SUCCESS:From 192.168.35.64 to 192.168.35.65 --(6の応答)

例 1: ACZ 停止時の AP メッセージ

ネットワークから孤立した ACZ は、発生しなかった。すなわち、本稿で提案する APP を使用することで、サーバを介さない P2P な AP 間ネットワークを構成でき、AP ネットワーク接続の頑健性が確認できた。

4.2 エージェントのコミュニケーションに関する評価

実験では、906 回のエージェントのコミュニケーションが試みられた。このうち、903 回のコミュニケーションが正常に行われた。残り 3 回については、相手のエージェントが存在しないため、コミュニケーションを行わないものであった。この実験で、エージェントは、相手の物理的位置を意識することなく、コミュニケーションが正常に行えた。このように APP を使用することで、ACZ はエージェントの位置透過性を与えることができた。

5 おわりに

本稿では、エージェントの活動を促進する環境 AP を相互運用するためのプロトコルとして APP を提案した。APP は、サーバを介さない P2P な AP 間ネットワークで、エージェントが物理的環境を意識することなくコミュニケーションを行うための環境を与える

ことができる。

この検証のため、我々の研究室で開発している ACZ に、APP を実装した結果、APP を実装した ACZ は、動的な経路変更機能を持つ P2P な ACZ ネットワークを構成することができた。

今後の課題は、ACZ 以外の AP に APP を実装し、ACZ と相互運用が可能であるかを検証することと、APP で提供されるエージェント検索方法の効率を検証することである。

参考文献

- [1] G. Zhong, K. Takahashi, T. Helmy, K. Takaki, T. Mine, S. Kusakabe and M. Amamiya, "KODAMA: As a Distributed Multi-agent System", Proceedings of the 7th International Conference on Parallel and Distributed Systems: Workshops, IEEE Computer Society Press, pp.435-440, Iwate, Japan, July.2000.
- [2] 高橋 健一, 高木 幸一郎, 鍾 国強, 雨宮 聡史, 峯 恒憲, 雨宮 真人, "分散マルチエージェントシステム KODAMA", 「ソフトウェアエージェントとその応用」特集ワークショップ, SAA2000, 157-164, 2000
- [3] T. Iwao, M. Takada, M. Amamiya, "Flexible Multi-Agent Collaboration using Pattern Directed Message Collaboration of Field Reactor Model", Approaches to Intelligent Agents - Second Pacific Rim International Workshop on Multi-Agents, LNAI, vol.1733, Springer, pp.1-15, 1999.
- [4] N. Minar, M. Gray, O. Roup, R. Krikorian and P. Maes, "Hive: Distributed Agents for Networking Things", IEEE Concurrency, vol.8, no.2, 2000.
- [5] FIPA : <http://www.fipa.org>
- [6] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language", In Software Agents, MIT Press, Cambridge, pp.291-316, 1997.
- [7] E.D. Pietro, A.L. Corte, O. Tomarchio, A. Puliafito, "Extending the MASIF Location Service into the MAP Agent System", IEEE Symposium on Computers and Communications, 2000.
- [8] D. Comer, D. Stevens (村井 純, 楠本 博之 (訳)), "TCP/IP によるネットワーク構築", 共立出版, 2001.

投機的計算を使用したマルチエージェントシステムでの問題解決手法

山本 京司 佐藤 健

In this paper, we propose a method of problem solving in multi-agent systems when communication between agents is not guaranteed. To solve a problem under incomplete communication environments, a method using speculative computation is proposed by Satoh et al. However, this method applies only in a case that another agent's responses to the agent's question can't be revised in solving a problem. In this paper, we extend speculative computation proposed by Satoh et al. to a case that responses can be revised.

1 はじめに

現在のマルチエージェントの研究の多くでは、エージェントが他エージェントに質問を与えた場合、質問を受けたエージェントから回答が来るまで、質問をしたエージェントの処理は中断されるのが普通である。しかし、この“処理が中断される”ということは、問題点となる。なぜなら、実際には、回答が来ない可能性という事も考えられるからである。例えば、インターネットのように通信が必ずしも保証されていない環境では、質問、回答が途中で失われる事が有りうる。このような事態が生じた場合に、処理が中断されるという事が大きな問題となるのは明らかである。

この様な問題を解決する方法として、投機的計算を

用いた手法が佐藤らによって提案されている[1] [2]. 佐藤らによって提案された手法は、あるエージェントから他エージェントに与えた質問の回答が翻らない場合において、上記の問題を解決している。しかし、各エージェントが投機的計算を行い、他エージェントに対して質問、回答を互いに送る様な場合には、回答が何回も変化することがあり得る。この様な場合、回答が翻らない場合に限られている佐藤らの手法では対応することが出来ない。

本論文は、各エージェントが投機的計算を行なうようなマルチエージェントシステムによる問題解決の手法を提案することを目的としたものである。その実現の為、あるエージェントからの回答が翻るような場合にも投機的計算を適用する手法の提案、及びここで提案した手法の計算機上への実装を行った。

本論文では、説明のため以下のような例を用いる。

- ある保険の契約を行う問題を考える。本社エージェント a 、営業エージェント b 、契約者 c がいる。
- a は、契約者 c がオプションをつけるかを営業エージェント b に聞く。
- b は、契約者 c に、オプションをつけるかどうかということを聞く。
- a は、 c がオプションをつけないならばプラン 1 を、オプションをつけるのであればプラン 2 を用意する。
- b は、 c からの回答を聞き、 a に回答する。

この例題で、以下の様な場合を考えてみる。

通常のマルチエージェントシステムの場合、エージェント a, b はそれぞれエージェント b 及び契約者

Keiji YAMAMOTO, 北海道大学大学院工学研究科,
Graduate School of Engineering, Hokkaido University

Ken SATOH, 国立情報学研究所, National Institution
of Information

c から回答が来なければ、処理を進めることはできない。しかし、ここでエージェント a が“多くの契約者はオプションを付けない”ということを知っていれば、 b からの回答を待たずに先行的に準備を進めることができる。

同様にエージェント b が“契約者 c は、多くの場合オプションを付ける”ということを知っていれば、事前にエージェント a に対してそのこと報告をすることができ、 a はその報告を受けて準備の修正を行なう。もしその後、 c から“オプションをつけない”という回答を b が受け取れば、 b は a に対して先程の報告を訂正し、さらに a は再度修正を行なう。

この場合、エージェント a と b がそれぞれ投機的計算を行なっている。もし回答が変化する場合に対応できなければ、エージェント b が先行的に出した回答をエージェント a に送ることは危険性が高い。なぜならば、エージェント b が出した回答は、投機的計算の性質上、保証されていないからである。

回答が変化する場合に対応するための手法の概略は、以下のようになる。

1. 各エージェントは事前に質問に対するデフォルトの仮回答を用意する。
2. 各エージェントは、用意した仮回答をもとに以降の処理を進める。
 - もしデフォルトの仮回答を使って計算を進められるならば、そのまま続ける。
 - またはデフォルトの仮回答とは矛盾する回答を使わなければ計算を進められなければ、その矛盾する回答の集合を覚えておき、計算は中断し、他の計算プロセスの処理を始める。(ここで、この集合を処理中断集合と呼ぶ。)
3. 計算中に回答が返されたならば、
 - (a) 現在計算中および中断されている全ての計算プロセスをチェックし、その中に回答と矛盾する仮定を使用しているものがあれば、その仮定を処理中断集合に移す。
 - (b) 処理が中断されている全ての計算プロセスの処理中断集合をチェックし、その中に回答と一致するものが含まれていれば、それを回答待ち集合から削除する。

4. 処理中断集合が空集合となっているものを選択して、計算を続ける。

4 では、処理中断集合が空集合の計算プロセスを選択している。これは、処理中断集合が空集合であるものは、デフォルトの仮回答または実際の回答と矛盾しておらず、処理を進めても成功する可能性が高いと思われるからである。

本論文では、他エージェントからの回答を待っているリテラルを集合として保存している。これにより、他のエージェントからの回答が変化する場合に投機的計算の導入を行うことができる。

2 投機的計算導入の実現

以下では、他エージェントからの回答が変化する場合に投機的計算を導入するための準備を行う。

2.1 枠組

ここでは、投機的計算の枠組の定義を述べる。

定義 1

Q を原子式としたとき、 Q を正リテラル、 $\sim Q$ を負リテラルとする。ここで、 $\sim$ は「失敗による否定」を表す。 Q をリテラルとしたときに、 $\sim \sim Q = Q$ と定義する。

定義 2

n 個のエージェントからなるマルチエージェントにおける投機的計算の枠組は、2 つ組 $\langle I, S \rangle$ である。ここで、 I は、エージェントの識別子の列 $\langle a_1, \dots, a_n \rangle$ であり、 S は、エージェントでの個別投機的計算の枠組の列 $\langle S_{a_1}, \dots, S_{a_n} \rangle$ を表し以下のように定義される。

各エージェントでの個別投機的計算の枠組 S は、 $\langle \Sigma, \mathcal{E}, \Delta, \mathcal{P} \rangle$ の 4 つ組であり、それぞれは以下の通りとする。

- Σ : 定数の有限集合、 Σ の要素をエージェント識別子と呼ぶ。
- $\mathcal{E}$: 外部述語と呼ばれる述語の集合、 Q が外部述語を持つリテラルで、 S がエージェント識別子のとき、 $Q @ S$ を質問リテラルと呼ぶ。 $\sim(Q @ S)$ を $(\sim Q) @ S$ と定義する。
- Δ : 以下の条件を満たす質問リテラルの基礎式
全ての外部述語 p を用いた基礎原子式 $p(t_1, \dots, t_n)$ と全てのエージェント S に対して、 Δ には

$p(t_1, \dots, t_n)@S$ または $(\sim p(t_1, \dots, t_n)@S)$ のいずれかが含まれているものとする。 Δ をデフォルト回答集合と呼ぶ。

- $\mathcal{P}$: 一般論理プログラム

質問リテラル $Q@S$ は、以下の二つの意味を持っている。

1. $\mathcal{P}$ のルール中の質問リテラル $Q@S$ は、 Σ 中のエージェント S に対して行なう質問を表す。
2. Δ における質問リテラルは、外部エージェント S への質問に対するデフォルトの仮回答を表す。もし $p(t_1, \dots, t_n)@S \in \Delta_{S'}$ ならば、質問 $p(t_1, \dots, t_n)$ に対する S のデフォルトの仮回答は yes であり、 $(\sim p(t_1, \dots, t_n))@S \in \Delta_{S'}$ ならば、質問 $\sim p(t_1, \dots, t_n)$ に対する S のデフォルトの仮回答は yes である。

先程の保険契約の例題をこの枠組を使用して表すと、以下のようになる。

枠組: $\langle \langle a, b \rangle, \langle S_a, S_b \rangle \rangle$

S_a

- $\Sigma_a = \{b\}$
 - $\mathcal{E}_a = \{op_c\}$
 - $\Delta_a = \{\sim op_c@b\}$
 - $\mathcal{P}_a$:
- $$prepare(P) \leftarrow plan(P).$$

$$plan(1) \leftarrow \sim op_c@b.$$

$$plan(2) \leftarrow op_c@b.$$

S_b

- $\Sigma_b = \{c\}$
 - $\mathcal{E}_b = \{op\}$
 - $\Delta_b = \{op@c\}$
 - $\mathcal{P}_b$:
- $$op_c \leftarrow op@c.$$

Δ_a は、“契約者はオプションをつけない” という仮定に、 Δ_b は “契約者 c はオプションをつける” という仮定に相当する。

2.2 準備

投機的計算の実行は、以下の二つの過程からなる。一つはプロセス簡約過程であり、もう一つは回答到着過程である。

プロセスは、(a) その計算を開始、または継続するかを判定する為の集合、(b) 計算の現在の状態、(c) そのプロセスで既につかわれたデフォルト、(d) 最初に与えられた質問の変数に対する解代入、(e) 回答を返す相手、の 5 つの状態からなる。

一つの質問に対する証明を行うために、一つのプロセスリストが作成される。プロセスは、場合分けのような選択肢がある場合に分岐し、各プロセスは、おのの別な分岐計算を表す。プロセス簡約過程では、計算を開始、または継続すると判定されたプロセスを新たなプロセスに変換する。プロセス簡約過程とは、各エージェントにおける通常のプログラムの処理に相当するものである。

一方、回答到着過程は、他エージェントからの回答が来たときの割り込み処理に相当するものである。本論文と [1] [2] との差異は、回答到着過程の手法の違いである。[1] [2] では、エージェントからの返答を待っている中断リテラルが 1 個だけであるのに対して、本論文では、中断リテラルを集合としてプロセス中に保存するという形にしている。そして、中断リテラルの集合が空集合ならば、そのプロセスは、他エージェントから送られた実際の回答、もしくはまだ回答が返されていないならば、事前に準備した仮回答と矛盾していないということになる。よって、そのプロセスを実行可能プロセスとし、また、そうでないときはプロセスを実行中断プロセスとしている。

定義 3

プロセスは 5 つ組 $\langle SGS, GS, AD, ANS, ID \rangle$ とする。ここで、

- SGS : 他エージェントからの回答を待っている GS 内の質問リテラルの集合。処理中断集合と呼ぶ。
- GS : 拡張リテラルの集合。ゴール列と呼ぶ。
- AD : 質問リテラルの集合。仮定デフォルトと呼ぶ。
- ANS : 初期ゴールにおける変数に対する解代入、または他エージェントから送られた質問リテラル。

- ID : 送信エージェント識別子.

SGS とは, 真偽値の値がデフォルトまたは実際に他エージェントから来た回答の値と異なる質問リテラルを格納する集合となる. この集合が空集合ならば, そのプロセスは現時点では仮定したデフォルト, または実際に来た回答とも矛盾していない. そのため, そのプロセスは成功する可能性が高く, 効果があると考えられるのでそのプロセスの計算を開始, または継続する. 空集合でなければ, そのプロセスは計算を行っても失敗する可能性が高いので, 中断して他のプロセスに実行を移す.

ANS は, 初期ゴールに変数が含まれている場合はその変数に対する代入が格納され, 他エージェントから質問を受け, その質問の証明を行う場合は, 質問リテラルが格納される.

ID は, **質問を送ったエージェントの識別子**である. エージェント S からエージェント S' への質問は $Q@S'$ from S という形で送られ, この場合, $ID = S$ となる. また, エージェントに対する質問がユーザーから与えられた場合は, $ID = u$ とする.

定義 4

プロセスにおいて, $SGS = \emptyset$ となるプロセスを実行可能プロセス, $SGS \neq \emptyset$ となるプロセスを実行中断プロセスとする.

計算状態を, 以下のリスト及び集合で定義する.

定義 5

- プロセスリスト PS とは, 実行可能プロセス, 実行中断プロセスの集合である.
- 既質問集合 AAQ とは, 他エージェントに送られた質問リテラルの集合である.
- 既回答集合 RF とは, 実際に他エージェントから返答された質問リテラルの集合である.

プロセスリスト PS は, 初期ゴールが与えられる度に新たに作成される. また, 一つのエージェント内で, ある初期ゴール GS から作成されたプロセスリストの処理は, 他の初期ゴール GS' から作成されたプロセス

リストの処理とは独立に行なわれる.

既質問集合 AAQ 及び既回答集合 RF は, 各エージェントに対して各々ひとつずつ作成される. あるエージェントにおいて, 複数のプロセスリストが作成された場合でも, 各々のプロセスリストから同じ AAQ を参照することにより, 質問の重複を避けることが可能となる.

ゴール G の証明について, 以下のように定義する.

定義 6

- ゴール G を証明する為のプロセスリスト中で, あるプロセスが $SGS = \emptyset$ かつ $GS = \emptyset$ となった場合, ゴール G の証明に成功したとみなす.
- ゴール G を証明するためのプロセスリスト中の全てのプロセスが中断された場合, すなわち全プロセスで $SGS \neq \emptyset$ となった場合, このゴール G の証明に失敗したとみなし, $\sim G$ とする.

証明の失敗の定義は, ゴール G を証明するための証明木において, 全ての枝に, 事前に用意した仮定デフォルトと異なるリテラルが現れるため, 失敗する可能性が高くなる. それ故, ゴール G の証明も失敗する可能性が高くなる. このため, ゴール G の証明を失敗したとみなし, $\sim G$ としている.

2.3 プロセス簡約過程

以下の簡約過程では, エージェント S において, 各ステップで新しく書き換わった PS_S, AAQ_S, RF_S をそれぞれ $NewPS_S, NewAAQ_S, NewRF_S$ と書くことにする. もし, 書き換えが書かれていない場合には, 変化しないことを表す. エージェント S に複数の初期ゴールが与えられた場合は, 以下の処理を各々の初期ゴールに対して独立に行なう.

初期ステップ: あるエージェント S で, 初期ゴール列 GS に対して実行可能プロセス $\langle \emptyset, GS, \emptyset, \emptyset, ID \rangle$ を証明手続きに与え, $AAQ_S = RF_S = \emptyset$ とし, 以下の**ステップ 1, ステップ 2, ステップ 3**を繰り返す.

ステップ 1: $\langle \emptyset, \emptyset, AD, ANS, ID \rangle$ というプロセスが

プロセスリスト PS_S 中に出現した場合、 $ID = S'$ ならば ANS をエージェント S' に返し、 $ID = u$ ならばユーザーに対し ANS を出力する。また、プロセスリスト中の全てのプロセスが実行中断プロセスとなった場合は、 ANS の否定を S' に返す、または u に出力する。

ステップ 2: さもなければ、プロセスリストから実行可能プロセス $\langle \emptyset, GS, AD, ANS, ID \rangle$ を選び、さらに、 GS から拡張リテラルを選び、 $PS' = PS - \{\langle \emptyset, GS, AD, ANS, ID \rangle\}$ および $GS' = GS - \{L\}$ とする。

ステップ 3: 上で選ばれたリテラル L に対して以下を行なう。

- L が正リテラルならば、
 $NewPS = PS' \cup$
 $\{\langle \emptyset, (\{body(R)\} \cup GS')\theta, AD, ANS, ID\theta \rangle |$
 $R \in \mathcal{P} \text{ and } \exists \text{ most general unifier(mgu)} \theta$
 $\text{s.t. } head(R)\theta = L\theta\}$
- L が基礎負リテラルならば、
 $NewPS = PS' \cup \{\langle \emptyset, NewGS, AD, ANS, ID \rangle\}$
ここで $NewGS =$
 $\{fail(body(R)) | R \in \mathcal{P} \text{ and } \exists \text{mgu } \theta$
 $\text{s.t. } head(R)\theta = L\theta\} \cup GS'$
- L が $fail(BS)$ という式ならば、
 - もし、 $BS = \emptyset$ ならば $NewPS = PS'$ 。
 - もし、 $BS \neq \emptyset$ ならば BS から B を選び、
 $BS' = BS - \{B\}$ とし、
* B が正リテラルならば、
 $NewPS = PS' \cup$
 $\{\langle \emptyset, NewGS \cup GS', AD, ANS, ID \rangle\}$
ここで $NewGS =$
 $\{fail((\{body(R)\} \cup BS')\theta) |$
 $R \in \mathcal{P} \text{ and } \exists \text{mgu } \theta \text{ s.t. } head(R)\theta =$
 $B\theta\}$
* B が基礎負リテラルまたは、
基礎質問リテラルならば、
 $NewPS = PS' \cup$
 $\{\langle \emptyset, \{\sim B\} \cup GS', AD, ANS, ID \rangle\} \cup$

$$\{\langle \emptyset, \{fail(BS')\} \cup GS', AD, ANS, ID \rangle\}$$

- L が基礎質問リテラル $Q@S'$ のとき
 - $L \notin AAQ$ かつ $\sim L \notin AAQ$ ならば、
エージェント S' に質問 $Q@S'$ from S を送り、
 $NewAAQ_S = AAQ_S \cup \{L\}$ とする。
 - $L \in RF_S$ ならば、
 $NewPS =$
 $PS' \cup \{\langle \emptyset, GS', AD \cup \{L\}, ANS, ID \rangle\}$ とする。
 - $\sim L \in RF_S$ ならば、
 $NewPS =$
 $PS' \cup \{\langle \{L\}, GS', AD \cup \{L\}, ANS, ID \rangle\}$ とする。
 - さもなければ $L \in \Delta$ ならば、
 $NewPS =$
 $PS' \cup \{\langle \emptyset, GS', AD \cup \{L\}, ANS, ID \rangle\}$
とする。
 - さもなければ $\sim L \in \Delta$ ならば、
 $NewPS = PS' \cup \{\langle \{L\}, GS', AD, ANS, ID \rangle\}$
とする。

2.4 回答到着過程

ここでは、回答が到着した場合についての処理手法を説明する。あるエージェント S' からエージェント S に対して回答 Q が返されたとき、 S は以下の処理を行う。以下の処理過程では、新しく書き換わった SGS_S, AD_S, RF_S をそれぞれ $NewSGS_S, NewAD_S, NewRF_S$ とする。もし、書き換えが書かれていなければ、変化しないことを表す。

- プロセスの処理
 PS 中の全てのプロセスに対して以下を行う。
 - $Q@S' \in SGS_S$ ならば、
 $NewSGS_S = SGS_S - \{Q@S'\}$ かつ
 $NewAD_S = AD_S \cup \{Q@S'\}$
 - $\sim Q@S' \in AD_S$ ならば、
 $NewSGS_S = SGS_S \cup \{\sim Q@S'\}$ かつ
 $NewAD_S = AD_S - \{\sim Q@S'\}$

上記の処理後、 $SGS_S = \emptyset$ となっているプロセス (実行可能プロセス) を選択し計算を進める。そう

でないプロセス (実行中断プロセス) は計算を中断する.

- RF_S の処理
 - $Q@S' \notin RF_S$ かつ $\sim Q@S' \notin RF_S$ ならば,
 $NewRF_S = RF_S \cup \{Q@S'\}$
 - $Q@S' \in RF_S$ ならば,
 $NewRF_S = RF_S$
 - $\sim Q@S' \in RF_S$ ならば,
 $NewRF_S = RF_S \cup \{Q@S'\} - \{\sim Q@S'\}$
 この処理は, 各エージェントからの最新の回答だけを蓄えておくということを表す.

3 計算機上への実装

提案した手法の計算機上への実装を行なった. システムの構成は図 1 のようになる.
 投機的計算部分に prolog を, 通信部分に Java を用い

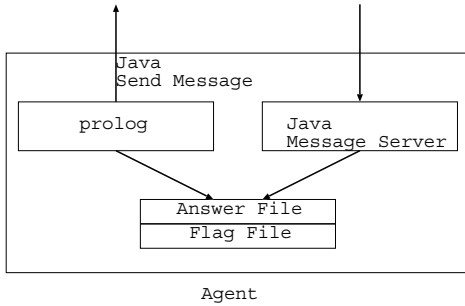


図 1 システム構成

て実装を行なった. 初期状態では, Flag File に “false” と書き込み, Answer File には何も書き込まないでおく.

prolog では, 投機的計算を進める過程において, 質問リテラルが出現した場合, またはゴールの証明に成功もしくは失敗した場合, 質問リテラルまたは回答を対象となるエージェントの Message Server に送信する.

一方, Java で作成した Message Server では, 他エージェントからのメッセージを受け取った場合, Flag File を “true” と書き換え, 実際のメッセージを Answer File に書き込む.

prolog では, 1 ステップ毎に Flag File のチェックを

行なう. この時, flag が “true” となっていれば, 他エージェントからのメッセージが来ているので, Answer File を読み込み, 投機的計算に他エージェントのメッセージを反映させる. その後, Flag File を “false” と書き換える.

保険契約の例題における全体の構成は, 図 2 の通りとなる. なお, c は投機的計算を行なうエージェントではなく, c からエージェント b への回答はユーザーが行なう.

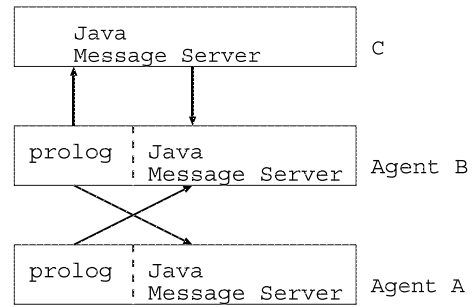


図 2 全体図

4 実行例

実行例として, 先に述べた保険契約の問題の実行履歴を示す.

以下で, エージェント a に $prepare(P)$ を与えた場合の実行履歴を示す.

最初に, エージェント a の履歴を示す.

1. $PS_a = \{\langle \emptyset, \{prepare(P)\}, \emptyset, \emptyset, \emptyset \rangle\}$,
 $AAQ = \emptyset, RF = \emptyset$
2. $PS_a = \{$
 $\langle \emptyset, \{plan(1)\}, \emptyset, \{P = 1\}, \emptyset \rangle,$
 $\langle \emptyset, \{plan(2)\}, \emptyset, \{P = 2\}, \emptyset \rangle$
3. $PS_a = \{$
 $\langle \emptyset, \{\sim op_c@b\}, \emptyset, \{P = 1\}, \emptyset \rangle,$
 $\langle \emptyset, \{plan(2)\}, \emptyset, \{P = 2\}, \emptyset \rangle$
4. $PS_a = \{\langle \emptyset, \emptyset, \{\sim op_c@b\}, \{P = 1\}, \emptyset \rangle,$
 $\langle \emptyset, \{plan(2)\}, \emptyset, \{P = 2\}, \emptyset \rangle$
 $AAQ_a = \{\sim op_c@b\}$
 エージェント a からエージェント b に
 $\sim op_c@b$ from a という質問を送る

ここで, a は b から回答が来る前に, デフォルト $\sim op_c@b$ を使用して $P = 1$ という解を得た.

つぎに, エージェント a から $\sim op_c@b$ from a という質問を受け取ったエージェント b の履歴を示す.

1. $PS_b = \{$
 $\langle \emptyset, \{\sim op_c\}, \emptyset, \{\sim op_c@b\}, a \rangle\},$
 $AAQ_b = \emptyset, RF_b = \emptyset$
2. $PS_b = \{$
 $\langle \emptyset, \{fail(op@c), \emptyset, \{\sim op_c@b\}, a \rangle\}$
3. $PS_b = \{$
 $\langle \emptyset, \{\sim op@c\}, \emptyset, \{\sim op_c@b\}, a \rangle\}$
4. $PS_b = \{$
 $\langle \{\sim op@c\}, \emptyset, \emptyset, \{\sim op_c@b\}, a \rangle\}$
 $AAQ_b = \{\sim op@c\}$
 エージェント b から c に対して, $\sim op@c$ from b
 という質問を送る

ここで, $\sim op@c$ は b のデフォルトと一致しないため, 処理中断集合に入り, この計算プロセスは中断される. また, $\sim op_c$ を証明するための計算プロセスが全て中断されたため, このゴールの証明に失敗したものとみなし, $\sim(\sim op_c)$, すなわち $op_c@b$ がエージェント a に対して回答される. 続いて, このエージェント b からの回答を受け取ったエージェント a の履歴を示す.

- (a) $PS_a = \{\langle \{\sim op_c@b\}, \emptyset, \emptyset, \{P = 1\}, \emptyset \rangle,$
 $\langle \emptyset, \{plan(2)\}, \emptyset, \{P = 2\}, \emptyset \rangle\}$
 $RF_a = \{op_c@b\}$
- (b) $PS_a = \{$
 $\langle \emptyset, \{plan(2)\}, \emptyset, \{P = 2\}, \emptyset \rangle,$
 $\langle \{\sim op_c@b\}, \emptyset, \emptyset, \{P = 1\}, \emptyset \rangle\}$
- (c) $PS_a = \{$
 $\langle \emptyset, \{op_c@b\}, \emptyset, \{P = 2\}, \emptyset \rangle,$
 $\langle \{\sim op_c@b\}, \emptyset, \emptyset, \{P = 1\}, \emptyset \rangle\}$
- (d) $PS_a = \{$
 $\langle \emptyset, \emptyset, \{op_c@b\}, \{P = 2\}, \emptyset \rangle,$
 $\langle \{\sim op_c@b\}, \emptyset, \emptyset, \{P = 1\}, \emptyset \rangle\}$

履歴 (a) において, b から回答 $op_c@b$ を受け取っ

たことによって, それまで計算を行っていたプロセス $\langle \emptyset, \emptyset, \{\sim op_c@b\}, \{P = 1\}, \emptyset \rangle$ 中の $\sim op_c@b$ が処理中断集合に移される. **このプロセスは処理中断集合が空集合でなくなるので, 実行中断プロセスとなり, 別のプロセスの計算が行なわれる.**

a が出力する解は $P = 2$ となる.

ここで, c からエージェント b に対して $\sim op@c$ という回答がきたとする. この時, エージェント b では以下のようになる.

1. $PS_b = \{$
 $\langle \emptyset, \emptyset, \{\sim op@c\}, \{\sim op_c@b\}, a \rangle\}$
 $RF_b = \{\sim op@c\}$
 先程まで SGS に入っていた $\sim op@c$ が AD に移された. これにより $SGS = \emptyset$ かつ $GS = \emptyset$ となるプロセスが出現し, $\sim op_c@b$ の証明が成功したとみなされ, エージェント a に回答される.
 エージェント a は, 先程とは翻った回答をエージェント b から受けることになる. この時のエージェント a は以下のようになる.

- (a) $PS_a = \{$
 $\langle \emptyset, \emptyset, \{\sim op_c@b\}, \{P = 1\}, \emptyset \rangle,$
 $\langle \{op_c@b\}, \emptyset, \emptyset, \{P = 2\}, \emptyset \rangle\}$
 $RF_a = \{\sim op_c@b\}$

エージェント a から, 再び解 $P = 1$ が得られる. この実行履歴で示した通り, エージェント a は翻った回答をエージェント b から受け取ることになるが, 最新の回答に基づいた解が得られる. また, この履歴では示していないが, エージェント b が c から翻った回答を受け取った場合でも, エージェント a の場合と同様にして最新の回答に基づいた解が得られる.

5 おわりに

本論文における成果は, 投機的計算の手法を拡張し, 他のエージェントからの回答が変化する場合に 대처することが出来る手法の提案を行なった事, またその手法を計算機上に実装し, 動作の確認を行なったことである. 今後の課題としては, 応用問題による効果の検証が挙げられる.

謝辞

本研究を進めるにあたり, 北海道大学の原口 誠先生, 大久保 好章先生, 神戸大学の井上 克己先生, 和歌山大学の坂間 千秋先生, 山梨大学の岩沼 宏治先生に貴重なご助言, ご指摘をいただきました。ここに謝意を示します。

参考文献

- [1] Satoh,K., Inoue,K., Iwanuma,K., Sakama,C., “Speculative Computation by Abduction under Incomplete Communication Environments”, *Proceedings of the Fourth International Conference on MultiAgent Systems*, to appear(2000)
- [2] 佐藤, 井上, 岩沼, 坂間, 不完全通信環境下におけるアブダクションによる投機的計算, 人工知能学会知識ベース研究会資料, SIG-KBS-9904, pp.43 - 48 (2000)
- [3] 山本, 佐藤, マルチエージェントシステムへの投機的計算導入手法, 第 15 回人工知能学会全国大会 (2001)

モバイルエージェント系における 分散制約充足

沼澤 政信 栗原 正仁 能登 正人

本論文では、モバイルエージェント系における分散制約充足について、いつ・どのような目的でエージェントの移動が考えられるのか基本的な考察をする。その中から、実ネットワーク中を流れる総通信量を抑制する目的で移動する場合に焦点を当て、簡単な計算モデルを構成し、計算機シミュレーションにより、このモデルに基づく移動が総通信量に与える効果について定量的な評価をする。分散制約充足アルゴリズムとしては分散 breakout アルゴリズムを仮定し、シミュレーションにより総通信量を測定した。その結果、以下のことがわかった。

1. 総通信量は、一般にリンク数の増加とともに急激に増加し、ピークに達した後急激に減少する。その後、緩やかに増加した後緩やかに減少する。
2. エージェントサイズが、上限として定義した s_n よりも十分小さいときには、移動したほうが総通信量が概ね削減できる。逆に、エージェントサイズが十分大きいときは、移動により総通信量はかえって増加する。

3. 総通信量がピークとなる困難な問題に対しては、エージェントサイズがある程度大きくても総通信量を抑制できる。

1 はじめに

近年のネットワーク技術の急速な発達により、コンピュータの利用形態は、単体で使用するというよりも、インターネットや LAN などのネットワークを介して使用する環境になった。そのような分散環境に対応した分散型ソフトウェアを柔軟に構築するためにエージェント技術が注目され、さらに最近ではそれを発展させたモバイルエージェントと呼ばれるホスト間を移動するエージェント技術が有効であると考えられている [1][2][3]。モバイルエージェントは、途中のタスクの内容を保持したまま別のホストに移動してタスクを継続することができるため、ネットワーク上のコンピュータ資源を有効利用することが可能である。よく知られているモバイルエージェントシステムとしては、General Magic 社による Telescript [4]、IBM による Aglets [5]、東芝による Plangent [6] などがある。

このような状況に対応して人工知能の分野においても、知的エージェントの実現基盤の一つとして知られている制約充足技術が、分散制約充足技術の研究に移行してきている [7]。そこでさらに、分散型ソフトウェアの構築を考慮し、分散制約充足の計算モデルもモバイルエージェントに対応するように拡張することは自然な研究の流れである。しかし、そのような研究例はまだみられない。エージェントが移動しようがしまいが、いわゆる位置透過性があるシステムにおい

Distributed Constraint Satisfaction for Mobile Agents
NUMAZAWA Masanobu, 小樽商科大学商学部社会情報
学科, Department of Information and Management
Science, Faculty of Commerce, Otaru University of
Commerce

KURIHARA Masahito, 北海道工業大学工学部情
報デザイン学科, Department of Information De-
sign, Faculty of Engineering, Hokkaido Institute of
Technology

NOTO Masato, 神奈川大学工学部電気電子情報工学科,
Department of Electrical, Electronics and Informa-
tion Engineering, Faculty of Engineering, Kanagawa
University

では、エージェント間のメッセージ交換の媒体が異なるだけで、メッセージの内容や交換のプロトコルが異なるわけではないので、抽象的あるいは論理的な意味では両者に差異はない。しかし、実装レベルでは、実ネットワーク中を流れる総通信量、延いては、実行効率の面で大きな違いとなりうる。

本論文ではそのような研究の第一歩として、まず、いつ・どのような目的でエージェントの移動が考えられるのか基本的な考察をし、その中から、実ネットワーク中を流れる総通信量を抑制する目的で移動するケースに焦点を当て、簡単な計算モデルを構成する。そして、計算機シミュレーションにより、このモデルに基づく移動が総通信量に与える効果について定量的な評価をする。

2 章では、モバイルエージェントと分散制約充足について概説し、その特徴を述べる。3 章では、考える種々の移動方式を簡単に考察した後、通信量を移動のトリガとする計算モデルを導入し、その評価のためのシミュレーション実験について述べる。4 章では、実験結果に基づき、この計算モデルの特徴と有効性を議論する。5 章では、まとめと今後の課題を述べる。

2 モバイルエージェントと分散制約充足

2.1 モバイルエージェントシステム

モバイルエージェントは、コンピュータ間の自律的移動能力を持つプログラムであり、エージェント自体が利用者の代理人としてネットワーク上を自律的に移動しながら特定のタスクを遂行する。そのため、利用者はネットワークに逐一接続して作業する必要がない。通常のエージェントでも、それぞれのホスト間で遠隔通信機能を使ってホスト間でのメッセージ通信が可能であるが、ネットワークが不安定で間欠的であったり、低速もしくは費用の高い通信路を使用しなければいけないといったケースが考えられるなど問題点も多い。なお、モバイルエージェントは AI 技術の関連で議論されている他のエージェント技術とは異なり、必ずしもインテリジェンスを持たないことが多いが、本研究で扱うモバイルエージェントは、分散制約充足という限定ながらも AI の基礎的な計算処理を能動的に実行し、移動先もそれ自身により選択し、移動する

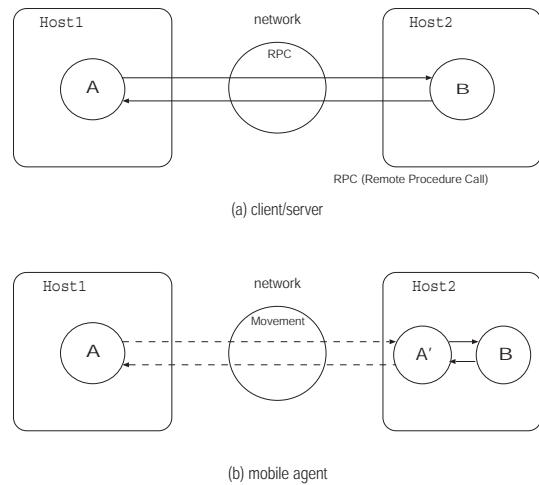


図 1 処理形態の違い

ことができるように設計されている。

モバイルエージェントを用いてコード転送し、それぞれのホストでその転送されたコードを実行するためには、セキュリティ保護、障害対策、トラフィック増加、管理の手間の増加といった問題が複雑化する。しかしながら、不安定で間欠的な通信路を常時使用する必要がなく、通信コストの削減が可能となるなどメリットが大きい。その理由は、従来のクライアント／サーバ方式とモバイルエージェント方式の二つの処理形態を比べることにより、容易に理解することができる。例えば、二つの異なるホストにいるエージェント間で 1000 回の要求を発行し、それに対する応答があるとき、クライアント／サーバ方式では、1000 回の要求と応答が Host1 と Host2 間のネットワーク上を往復することになる (図 1(a))。一方、モバイルエージェント方式 (図 1(b)) では、エージェントが 1 往復するだけで 1000 回の要求と応答は、Host2 上でローカルに行うだけで済む。この 1000 回が、非常に費用の高い通信路であったり、要求の回数が 1 万回、10 万回と増えていった場合を想定すると、移動して処理を行う効果は高いと想像できる。

また、この処理形態の違いにより、もう一つ重要な結果が生じる。クライアント／サーバ方式では、毎回の要求と応答がネットワーク上を行き来するため、ホスト間のコネクションは、常時維持しなければならない

いのに対して、モバイルエージェント方式では、エージェントが移動するときだけコネクションを確立すればよい。これは、電話回線でモデムを利用する場合の通信や、無線を利用しているためコネクションが不安定な携帯電話、移動体通信などの通信形態にとっては非常に有利である。

2.2 分散制約充足

制約充足問題 (Constraint Satisfaction Problem: CSP) [8] は、離散値をとるいくつかの変数に割り当てられる値の組合せのうち、与えられた制約をすべて満たす組合せを発見する探索問題である。分散制約充足問題 [9][10] は、制約充足問題の変数と制約が複数のエージェントに分散された問題とみなすことができ、さまざまな問題が分散制約充足問題として定式化されている [11][12][13][14]。制約充足問題では、初期状態において一つのアルゴリズムが全体の問題を見渡すことができるのに対し、分散制約充足問題の特徴は、初期状態はもちろんのこと、任意の時点においてそれぞれのエージェントは全体の状態を知らないことである。このため、各エージェントがどのような手順で、どのような情報を交換し合えば、全体として制約を満たす解が発見できるかが問題となる。

制約充足問題を解くアルゴリズムは、厳密解法である木探索アルゴリズムおよび近似解法である反復改善型アルゴリズムの二つに大きく分類される [15]。前者は整合性をとりながら部分解を拡張して探索木をたどり完全な解を求める。後者は変数に制約を満足しない値を割り当てた状態から制約違反を局所的に改善していくものであり、横尾、平山 [16] によって提案された反復改善型分散制約充足アルゴリズム (分散 breakout) が効果的な状態空間探索法の一つとして注目されている。

3 モバイルエージェント間の分散制約充足

本章では、モバイルエージェントと分散制約充足の二つの技術を結びつけてモバイルエージェント間での制約充足方式を提案する。はじめに、考えうるいくつかの移動方式を列挙し、次に、そのうちの一つである通信量に基づく移動方式を詳細化した計算モデル

を構築した後、それを評価するためのシミュレーション実験の条件を設定する。

3.1 種々の移動方式

現在、モバイルエージェントを分散制約充足に導入した例はまだないので、研究の第一歩として、いつ・どのような目的で移動したらよいのかを検討する必要がある。以下に、タイミングと移動の目的として考えられる基本的な移動方式を述べる。

1. 時間による移動

一定時間、他のエージェントから何のメッセージ応答もないときに移動する。現在エージェントが存在しているホストが過負荷である、または何らかの理由でプラットフォームの状態が不具合であるかもしれないと考えられるときに移動するため、本移動方式は、ネットワークで問題を解くときのロバスト性を高める効果が期待できると考えられる。

2. 準局所最適時の移動

エージェントが準局所最適状態になったときに移動する。準局所最適の状態を抜け出すには多くのメッセージ交換が必要になる場合がある。本移動方式は、そのような場合、その負荷を軽減し得る可能性を持つ。

3. メタレベルからの移動要求による移動

アプリケーションに依存して、自分が求めるエージェントに連絡をとり移動する。本移動方式は、タスクとして分散制約充足だけを行う研究レベルのエージェントシステムには付加されるものではない。しかし、このタスク機能を一部として包含する現実のアプリケーションにおいては「メタレベル」での要求によって移動する必要があることもあり得ると考えられる。

4. 計算負荷による移動

自分が存在しているホストマシンの負荷が所定の値を超えたときに移動する。本移動方式では、資源の効率的な利用が期待できると考えられる。

5. 通信量による移動

メッセージの累積量が所定の量を超えたときに、一方のエージェントが他方のエージェントの

いるホストマシンへ移動する。本移動方式は、非常に汎用的な単純な方法であり、通信コストの削減が期待できると考えられる。

以下では、「通信量による移動」に基づく考え方で具体的に計算モデルを構築し、基盤となる分散制約充足アルゴリズムとして分散 breakout アルゴリズム [16] を採用してシミュレーションを実施し、移動が通信量に与える効果について評価する。

3.2 通信量をしきい値とする移動方式

任意の分散制約充足アルゴリズムが与えられたとする。そのアルゴリズムは、必ずエージェントが他のエージェントとメッセージ交換しながら制約充足のための計算を実行しているようにモデル化されていると仮定する。我々は、各エージェント i が現在自分のいるホストと異なるホストにいるエージェント j とメッセージ交換（送信、受信）する部分のプログラムコードに手を入れ、エージェントにこれまでの通信量の累積を把握させる。そして、それがあしきい値を超えれば、エージェント i が他のホストに移動して、その後本来の計算を続行するような計算モデルを構築したい。そのような制限の下でもさらにいくつかのバリエーションが考えられるが、本論文ではこれから述べるような最も単純なものの一つを考察の対象としたい。

エージェント間で交換するメッセージ 1 つあたりのサイズ（メッセージサイズ）は簡単のため固定されていると仮定し、それを 1 と正規化して通信量の単位とする。エージェント i について、

q_{ij} : エージェント j とのメッセージ累積量

Q_i : 総メッセージ累積量

C_i : 総通信量

t_i : しきい値

とそれぞれ表し、初期値の設定は、 $q_{ij} \leftarrow 0$, $Q_i \leftarrow 0$, $C_i \leftarrow 0$, $t_i \leftarrow$ エージェントサイズとする。エージェントサイズは、エージェントの大きさ（すなわち、プログラムコードおよびエージェントの状態を表すデータの量）であり、エージェントが一回移動するのにかかる通信量でもある。簡単のために、これも固定されていると仮定する。移動条件の初期しきい値は、エー

ジェントが移動するための初期値である。直観的に、メッセージ通信量がエージェントサイズより相対的に小さいうちは移動することによってかえって大きな通信量の増加をまねくので、しきい値の考えうる最小の値という意味で初期値をエージェントサイズと等しくした。以上の準備のもとで、エージェント i がエージェント j とメッセージ交換したときに実行すべき処理は以下の手順に従う。

1. $q_{ij} \leftarrow q_{ij} +$ メッセージサイズ;
2. $Q_i \leftarrow Q_i +$ メッセージサイズ;
3. $C_i \leftarrow C_i +$ メッセージサイズ;
4. もし、 $Q_i > t_i$ ならば以下の処理 (4.1~4.4) をする;
 - 4.1 $j \leftarrow \max\{q_{ij}\}$ を与える j ;
 - 4.2 エージェント j のいるホストへ移動する;
 - 4.3 $C_i \leftarrow C_i +$ エージェントサイズ;
 - 4.4 $t_i \leftarrow 2t_i$;
5. 分散制約充足アルゴリズムの処理;

最初に、しきい値を満たす前は、各エージェント間でメッセージ交換を行いながら、メッセージサイズを q_{ij} , Q_i , C_i に累積していく。その後、メッセージ累積量がしきい値を超えたときに、エージェントは、メッセージ交換したすべてのエージェントの中で個々のメッセージ量が最も多いエージェントのいるホストに移動する。このとき、総通信量にエージェントサイズを累積する。なお、メッセージ量の最大値をもつエージェントが複数の場合は、その中から非決定的に選択し移動先を決める。しきい値は移動するたびに 2 倍にしていく。これは、データ構造の分野で配列やハッシュ表のサイズを動的に拡張する際にも用いられる考え方の一つとほぼ同様で、単純ながらも自然な考えである。

本研究では、基盤となる分散制約充足アルゴリズムとして分散 breakout アルゴリズムを採用するが、システムを実ネットワーク上に実装するのではなく、性能の事前評価を目的とし、1 台のマシン上でのシミュ

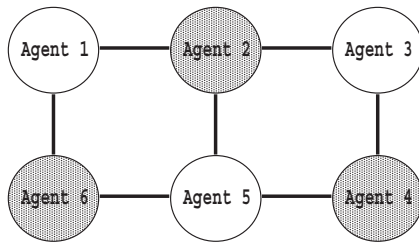


図 2 分散グラフ色塗り問題の例

レーション・システムとして実現している。

3.3 シミュレーション実験

分散制約充足問題の代表的な例題として分散グラフ色塗り問題がある。グラフ色塗り問題とは、任意のグラフと色数が与えられ、グラフ上の隣り合うノードを異なる色で塗るものである。このグラフのノードをエージェントとみなすことで、分散グラフ色塗り問題のエージェントネットワークが得られる。色数 2, エージェント数 6, リンク数 7 の分散グラフ色塗り問題の例および解を図 2 に示す。

3 色の分散グラフ色塗り問題をランダムに発生させ、メッセージサイズを 1 として表 1 に示す条件で、通信量に関する移動のシミュレーション実験を行う。

表 1 実験内容

	エージェント数	リンク数	エージェントサイズ
実験 1	30	30~270	10, 70, 300, ∞
実験 2	60	60~1100	50, 300, 500, ∞
実験 3	90	90~2600	100, 300, 500, ∞
実験 4	120	150~4500	100, 400, 600, ∞
実験 5	150	150~7000	100, 500, 900, ∞

実験では、リンク密度 0~100% の範囲でリンクの本数をパラメータとして設定し、各エージェント数 (30~150) に対してリンク数を変化させていき、各エージェントサイズでの通信量をそれぞれ求める。リンク数ごとに 100 回ずつ問題を解き、そのときの平均値をとる。なお、この問題では、リンク密度 25% 前後付近の値のところで最も難しい問題が存在することが知られている [17][18]。便宜上、エージェントを移動させない (固定した) ケースをエージェントサイ

ズ ∞ で表す。

4 実験結果と考察

シミュレーション実験の結果として実験 1, 実験 3 および実験 5 (エージェント数が 30, 90 および 150) のときの総通信量のグラフをそれぞれ図 3 ~ 図 5 に示す。

はじめに、エージェント数が十分多い実験 (図 4, 図 5) において、比較の基準となるエージェントサイズが ∞ (すなわち移動しないケース) のグラフの特徴を確認する。リンク数が増加していくと、ある狭い範囲で急激に総通信量が増加し、そのピークを超えると急激に減少している。これはいわゆる「相転移」などと呼ばれる現象としてよく知られている [19]。その後、総通信量は緩やかに増加に転じた後、再度、緩やかに減少する。

次に、このグラフを他の (移動する場合の) エージェントサイズのケースと比べてみる。エージェントサイズが十分小さい場合には、リンク数にかかわらず、移動しない (固定した) ときの総通信量よりも移動したときの総通信量の方が概ね削減できていることがわかる。逆に、直観的にも明らかなことだが、エージェントサイズがあまりにも大きい場合は、移動することによりその移動に必要な大きな通信量が加算されるため、総通信量は移動しないときよりもかえって大きくなってしまう。さらに、重要なこととして、リンク数が 250 (図 4) 付近 (図 5 では 400 付近) の最も難しい問題による通信量のピークが、いずれのエージェントサイズの場合も、 ∞ のケースより低く抑えられていることがわかる。これは、今回設定した移動基準が、過大な通信量を抑制するように働くことによるものと考えられる。

エージェントサイズがある程度小さければ移動の効果があり、逆にある程度を超えて大きければ移動は逆効果となることがわかった。そこで、移動が効果的でありうるようなエージェントサイズの上限を求めてみる。エージェント数に加えてリンク数が既知として与えられたときは、エージェントサイズをパラメータとして今回のような実験をして、移動しないケースと比較すれば上限は自明な方法で実験的に求めら

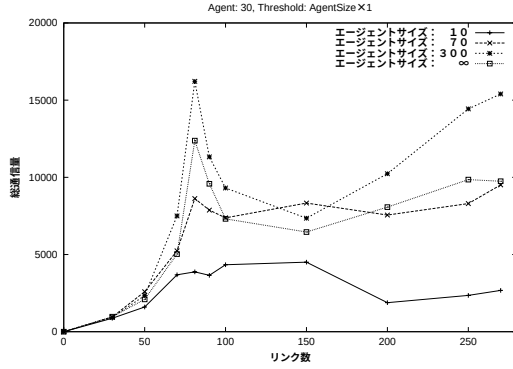


図 3 実験 1 (エージェント数: 30)

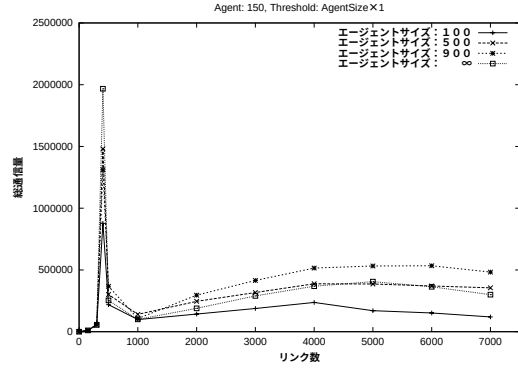


図 5 実験 5 (エージェント数: 150)

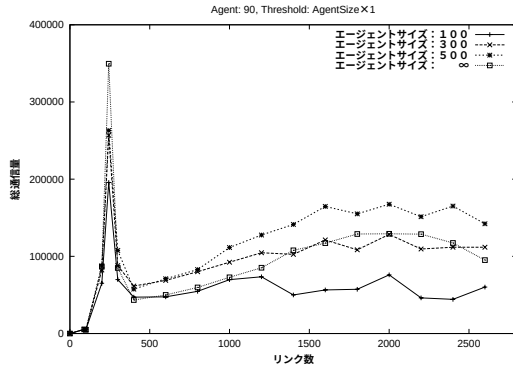


図 4 実験 3 (エージェント数: 90)

ただし, $L = n(n-1)/2$ はリンク数の上限値である. 今回の実験から求めた $D_{n,s}$ の概算値の一部を表 2 に示す.

 表 2 $D_{n,s}$ の計算結果

s	エージェント数 n				
	30	60	90	120	150
10	-4220	-21800	-61500	—	—
50	-1520	-14500	-51400	-124000	—
100	1180	-7270	-38600	-107000	—
200	3230	5640	-16700	-71900	-94200
300	—	14000	2730	-35100	-59400
400	—	18500	18400	-5380	-28900
500	—	—	29500	26200	17300

れる. しかし, 分散環境においては一般にエージェントは問題の全体を知らないで, エージェント数のみは知っているとして, リンク数を知らない場合の上限を定義してその概数を試算しておくことが興味深い. そのようなヒューリスティクスはあまり厳密に論じても意味が薄いので, 本論文では非常に単純なものだけを以下に述べる.

はじめに, エージェント数が n , エージェントサイズが s のときの総通信量をリンク数 ℓ の関数として $C_{n,s}(\ell)$ で表す (図 3 ~ 図 5 の一本一本の折れ線がそのような関数の実例を表している). そこで, $s = \infty$ の関数との平均的な差を以下の式で定める.

$$D_{n,s} = \frac{1}{L} \sum_{\ell=1}^L [C_{n,s}(\ell) - C_{n,\infty}(\ell)]$$

表 2 からわかるように, エージェント数を固定したとき, エージェントサイズ s が十分小さいときは $D_{n,s} < 0$ であることから, 移動が効果的であるといえる. 逆に, s が十分大きいときは $D_{n,s} > 0$ であることから, 移動は効果的でないといえる. そこで, 先ほど論じた上限として, $D_{n,s} < 0$ を満たす最大の s の値を採用する. その値は n 毎に定まることから, s_n と表すことにする. 表 2 から補間によって求めた s_n の近似値を表 3 および図 6 に示す.

 表 3 s_n の近似値

n	30	60	90	120	150
s_n	75	156	286	417	463

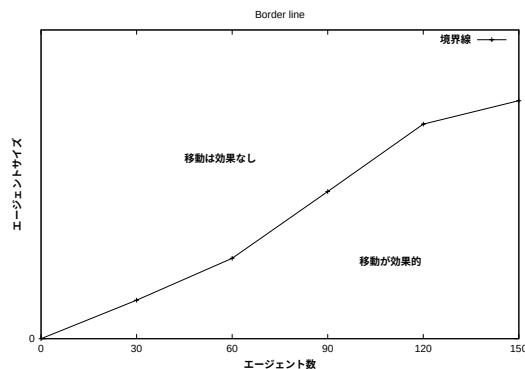


図 6 移動の境界線

きわめて大まかな指針として、リンク数が未知の場合、メッセージサイズを 1 としたときのエージェントサイズが、エージェント数のおよそ 3 倍程度以内のときは、移動による通信量の削減が期待できる。エージェントを実装するクラスファイルを各ホストに事前に置いておくなど、実装上の工夫でこの条件を満たせば、通信量の削減という理由だけで、エージェントを移動させる正当な理由となり得る。

ただし、エージェントサイズがこの指針より大きいときに本方式が役に立たないというわけではない。リンク数が既知で、特に、計算量的に「難しい問題」を発生させるようなリンク数のときには、多少エージェントサイズが大きくても、それを大きく上回るメッセージ量を削減できるので効果が大きい。また、3.1 節で述べたような種々の理由で移動そのものが効果を発揮することもあり得る。

5 おわりに

本研究では、モバイルエージェント系における分散制約充足について、いつ・どのような目的でエージェントの移動が考えられるのか基本的な考察をし、その中から、総通信量を抑制する目的で移動するケースに焦点を当て、簡単な計算モデルを構成した。次に、分散制約充足アルゴリズムとして分散 breakout アルゴリズムを仮定し、シミュレーションにより総通信量を測定した。その結果、以下のことが確認できた。

1. 総通信量は、一般にリンク数の増加とともに急

激に増加し、ピークに達した後急激に減少する。その後、緩やかに増加した後に緩やかに減少する。

2. エージェントサイズが、上限として定義した s_n よりも十分小さいときには、移動したほうが総通信量が概ね削減できる。逆にエージェントサイズが十分大きいときは、移動により総通信量はかえって増加する。
3. 総通信量がピークとなる困難な問題に対しては、エージェントサイズがある程度大きくても総通信量を抑制できる。

今後の課題は以下のとおりである。

- 分散 breakout アルゴリズム以外の分散アルゴリズムの場合について検討する。
- 3.1 節で示したような通信量以外の移動方法について検討する。
- 1 つのホストに多数のエージェントが集中すると、CPU タイムが各エージェントに分配されて 1 エージェント当りの計算スピードが減少する問題があるので、それを解決するモデルを考えたい。つまり、上記の 2 つの検討事項を踏まえた上で、複合的要因により移動を決定する方法の検討を行う。
- 実際のモバイルエージェント系に実装を行い、シミュレーションとの比較を行う。

謝辞 本研究の一部は、文部科学省科学研究費（課題番号 12780246）の補助によって行われた。

参考文献

- [1] 本位田真一, 飯島正, 大須賀昭彦: エージェント技術, 共立出版 (1999).
- [2] 長尾確編著: エージェントテクノロジー最前線, 共立出版 (2000).
- [3] 新谷虎松, 大園忠親, 福田直樹: モバイルエージェントの応用—マルチエージェントシステムのためのモビリティの利用—, 人工知能学会誌, Vol. 16, No. 4, pp. 488–493 (2001).
- [4] White, J. E.: *Mobile Agents, Software Agents* (Bradshaw, J. M.(ed.)), AAAI Press/The MIT Press, chapter 19, pp. 437–472 (1997).
- [5] Lange, D. B. and Ohshima, M.: *Programming and Deploying Java Mobile Agents with Aglets*, Addison-Wesley (1998).

- [6] Ohsuga, A., Nagai, Y. and Irie, Y.: Plangent: An Approach to Making Mobile Agents Intelligent, *IEEE Internet Computing*, Vol. 1, No. 4, pp. 50-57 (1997).
- [7] 横尾真, 平山勝敏: CSP の新しい展開: 分散/動的/不完全 CSP, 人工知能学会誌, Vol. 12, No. 3, pp. 33-41 (1997).
- [8] Mackworth, A. K.: Constraint Satisfaction, *Encyclopedia of Artificial Intelligence* (Shapiro, S. C.(ed.)), Wiley-Interscience Publication, New York, pp. 285-293 (1992).
- [9] 平山勝敏, 横尾真: 分散不完全制約充足問題, 人工知能学会誌, Vol. 14, No. 4, pp. 60-69 (1999).
- [10] Yokoo, M. and Hirayama, K.: Algorithms for Distributed Constraint Satisfaction: A Review, *Autonomous Agents and Multi-Agent Systems*, Vol. 3, No. 2, pp. 198-212 (2000).
- [11] Conry, S. E., Kuwabara, K., Lesser, V. R. and Meyer, R. A.: Multistage Negotiation for Distributed Constraint Satisfaction, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 6, pp. 1462-1477 (1991).
- [12] Huhns, M. N. and Bridgeland, D. M.: Multi-agent Truth Maintenance, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 6, pp. 1437-1445 (1991).
- [13] Lesser, V. R. and Corkill, D. D.: The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks, *AI Magazine*, Vol. 4, No. 3, pp. 15-33 (1983).
- [14] Sycara, K. P., Roth, S., Sandeh, N. and Fox, M.: Distributed Constrained Heuristic Search, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 6, pp. 1446-1461 (1991).
- [15] Morris, P.: The Breakout Method for Escaping from Local Minimum, *Proc. of the Eleventh National Conference on Artificial Intelligence*, pp. 40-45 (1993).
- [16] 横尾真, 平山勝敏: 分散 breakout : 反復改善型分散制約充足アルゴリズム, 情報処理学会論文誌, Vol. 39, No. 6, pp. 1889-1897 (1998).
- [17] Cheeseman, P., Kanefsky, B. and Taylor, W.: Where the Really Hard Problems Are, *Proc. of 12th International Joint Conference on Artificial Intelligence*, pp. 331-337 (1991).
- [18] Minton, S., Johnston, M. D., Philips, A. B. and Laird, P.: Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems, *Artificial Intelligence*, Vol. 58, pp. 161-205 (1992).
- [19] Hogg, T., Huberman, B. A. and Williams, C. P.: Phase Transitions and the Search Problem, *Artificial Intelligence*, Vol. 81, pp. 1-15 (1996).

ユーザ選好に基づく分散コンテンツキャッシュポリシー

中塚 康介 八槇 博史

Abstract.

To deliver multimedia contents efficiently, a content delivery network (CDN) is used in the Internet. CDN has caches in points-of-presence of internet service providers and stores multimedia contents actively. In CDN, it is important to choose contents to cache, because the caching policy is important to effectiveness.

In this paper, we formulate the preference of end-users and describe the mechanism for helping constructing caching policy. Our approach is based on market mechanism where producer agents and consumer agents negotiate and trade resource. We believe that it not only reaches effective resource allocation but also matches with business model for CDN.

1 はじめに

現在, web が普及し, その上でマルチメディアコンテンツが流通しているが, 音声や動画などのマルチメディアコンテンツは, 帯域幅などのネットワーク資源を多く使用し, ネットワークや web サーバに大きな負荷をもたらすことが問題となっている. この問題の解決のために, インターネットサービスプロバイダ

構内にキャッシュを設置し, コンテンツのキャッシュを行うことによって, ネットワークや web サーバへの負荷を減少させるコンテンツデリバリネットワーク (Content Delivery Network; CDN) が利用されている.

コンテンツデリバリネットワークでは, 各キャッシュにどのようなコンテンツをキャッシュするかは, コンテンツ配信者の決定に委ねられている. 現在コンテンツデリバリネットワークは大規模なコンサートのライブ中継や, 社内教育などの一定の利用が見込まれるコンテンツに対して適用されており効果を上げているが, キャッシュコンテンツの選択にユーザの個別の選好が集約できず, エンドユーザの要求に応じてキャッシュするコンテンツを選択・配信し, コストを回収するモデルが求められている.

本稿では, この問題に対して, エンドユーザの潜在的な選好 (preference) のモデルを構築し, 収集したエンドユーザの選好を反映しながらコンテンツ配信に対する無駄のないネットワーク資源の利用, 及び, ネットワーク事業者, コンテンツ配信者, エンドユーザのコストの配分への応用を考えたキャッシュポリシーを提案する. これにより, 効率的なネットワーク資源の利用をはかるとともに, コンテンツ配信におけるビジネスモデルへの応用が考えられコンテンツ配信に貢献するものであると考えている.

本稿で提案するモデルは, これまで研究・開発が行われている分散キャッシュ技術により構築されたコンテンツデリバリネットワークの中で, ネットワーク事業者, コンテンツ配信者, 及び, エンドユーザの各

Kosuke NAKATSUKA, 京都大学 大学院 情報学研究科 社会情報学専攻, Department of Social Informatics, Kyoto University

Hirofumi YAMAKI, 京都大学 大学院 情報学研究科 社会情報学専攻, Department of Social Informatics, Kyoto University

主体のコスト負担, 特にエンドユーザに対する課金を含めて, キャッシュの置くコンテンツの選択を支援することを目標としており, 従来のキャッシュ技術と共に, コンテンツデリバリネットワークの効率的, かつ, 持続的な運用を図るものである.

2 コンテンツデリバリネットワーク

コンテンツデリバリネットワークは, 図 1 を一例とする構造を持つ仮想的なネットワークである.

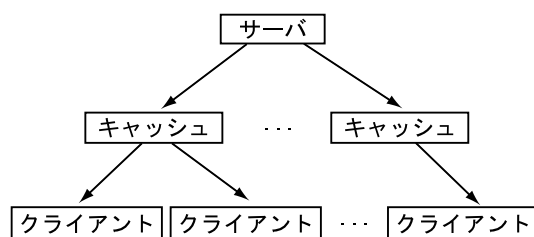


図 1 CDN のアーキテクチャ

図 1 において, サーバはテキスト文書, 画像, 音声, 動画などのコンテンツが保存されるホストであり, クライアントは, コンテンツを利用するエンドユーザを表している.

従来のクライアント/サーバモデルでは, サーバにクライアントが直接要求を出してデータを入手するものであり, サーバの負荷が高くなる, また, ネットワークにより大きな帯域幅を要するなどの問題があった.

これに対して, CDN では, サーバとクライアントの間にキャッシュを導入することにより, 負荷をサーバとキャッシュで分散し, 必要な帯域幅を減少させている. CDN は, 以下のように動作する.

1. インターネットサービスプロバイダの POP^{†1}にキャッシュを設置する.
2. キャッシュは, サーバからデータをコピーし, クライアントからの要求に備える. どのデータをコピーするかは, 管理者が設定として与える.
3. クライアントからの要求はサーバに対して行わ

れるが, 実際にはその要求は, 経路上に存在するキャッシュが途中で横取り (intercept) する. 要求を横取りしたキャッシュは, “どこからのアクセスか”, “何を要求しているか” を調べ, ユーザから近く負荷の低いサーバへの要求振り分けや, データがキャッシュされている場合は, サーバにかわり, そのデータの送信を行う.

クライアントからの要求は経路上で横取りされるため, web ブラウザなどを変更する必要はなく, 透過的に CDN を用いることができる.

従来 CDN に類似する技術としてプロキシ型のキャッシュサーバが用いられてきたが, ユーザからの要求によってキャッシュがはじめて行われ, キャッシュを行うポリシーをサービス提供者が設定することができないという問題や,

- 要求は多くのコンテンツに対して行われ, 一部の人気のあるコンテンツのみをキャッシュすることは, ある程度有効ではあるが, 十分ではない
- web ページのアクセス頻度とページサイズには弱い相関しかない
- web ページのアクセス頻度と更新頻度には弱い相関しかない

といった問題[1] から, ポリシ設定が困難であり, また, 有効性にも限界がある. これに対して, コンテンツデリバリネットワークでは, サーバとクライアントの間にキャッシュを導入し, このキャッシュにサーバから特定のコンテンツを前もってコピーしておき, クライアントからの要求に備えることで, 負荷を分散し, 配信に必要な資源を減少させている.

2.1 CDN の利点

CDN は, インターネットサービスプロバイダによる配信など, コンテンツ配信者が明らかであり, コンテンツ利用者が限られている, あるいは, コンテンツ利用が十分に見込まれる場合のように, 単一のポリシーが定義できる場合の負荷分散・高速配信に有効である. このような条件を満たすものとして, 各地に支社のある企業での企業内教育や, コンサートのライブ配信など, 特定のイベントにおけるストリーミング配信などがあり, 負荷分散が実現されている.

^{†1} Point-of-Presence, サービスプロバイダへのアクセスポイントを意味する

2.2 CDN における問題

CDN では、どのコンテンツをどのキャッシュにコピーするかを決定する必要がある。現在の CDN では、キャッシュ技術については、[2] にあげられるようなキャッシュアルゴリズムが適用され、効果を上げている。一方で、こういったコンテンツをコピーするかのポリシーについては、CDN プロバイダがコンテンツプロバイダと契約し、その契約に基づいてコンテンツを優先的にキャッシュするようになっており、各エンドユーザの要求に応じてコンテンツを配信し、配信についてのコンテンツやネットワークのコストを回収するモデルとはなっていない。このため、

- コンテンツプロバイダにとっては、コンテンツプロバイダが提供するコンテンツに対し、エンドユーザの利用を見通して、CDN を運用する必要があること、
- CDN プロバイダにとっては、ネットワークや計算機等の性能の向上に対して配信設備を維持し、継続して事業を行う必要があること、
- エンドユーザにとっては、高速なコンテンツ配信が望ましい

といった各主体の要求を満たすようなビジネスモデルの構築とそれに対応したキャッシュポリシーが必要となっている。この問題に対し、エンドユーザの嗜好を集約し

- エンドユーザ、コンテンツプロバイダ、ネットワークプロバイダの各主体にとって効率的なコンテンツ選択
- 各主体に対するコスト負担の決定

の 2 点が可能なポリシーの構築を目標として、市場機構による最適化手法を適用したキャッシュコンテンツ決定手法を提案する。

3 キャッシュコンテンツ決定プロトコル

キャッシュポリシーは、エンドユーザとキャッシュ間で効用情報をやりとりするプロトコル、及び、得られた効用情報をもとに適切なコンテンツを決定するアルゴリズムからなる。本章では、キャッシュポリシーを運用するためのエンドユーザとキャッシュの間のプロトコル記述する。

エンドユーザは、図 2 に示すように最寄りのインターネットサービスプロバイダの POP に設置されたキャッシュに対して接続されており、サーバに対する要求は、このキャッシュを経由するものとする。

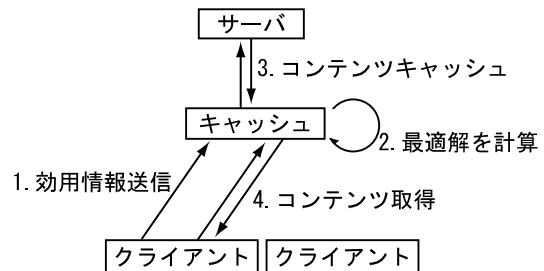


図 2 エンドユーザ・キャッシュ間プロトコル

以下、キャッシュコンテンツ決定のプロトコルを記述する。

1. エンドユーザはどのコンテンツがどの程度望ましいのかを表す効用情報を効用関数として送信する。効用関数はコンテンツに対して十分小さく、コンテンツ配信そのものに影響を与えないと考えられる。
2. キャッシュは、キャッシュの CPU 能力・ストレージ・ネットワークの帯域幅などの制約条件の中で集められた効用関数を最大化するようなコンテンツ集合を計算する。
3. キャッシュは得られたコンテンツ集合のコンテンツをキャッシュプロトコルに従い取得する。
4. エンドユーザはコンテンツに対する要求を行いコンテンツを取得する。

以上のプロトコルにより、クライアントの効用情報が収集され、最適なコンテンツ集合が決定される。

4 キャッシュ・エンドユーザ間のキャッシュ決定アルゴリズム

前節のように集められた効用情報は、映画よりも音楽が望ましいといったコンテンツ種別毎の嗜好が記述されているが、ストレージや帯域幅など資源に制約がある中で、このような嗜好を反映した資源割当てが可能な手法に経済学の知見を応用した市場機構による

資源割当てがある [3]. 市場機構による資源割当ては、ストレージや帯域幅など資源が限られた状況でも、利用者間の要求の衝突を調停しながら利用者の選好に応じた資源の分配を可能にする最適化手法であり、ネットワークの QoS 制御などに適用されている [4].

コンテンツデリバリネットワークでは、ネットワーク事業者、コンテンツ提供者、エンドユーザの各主体が存在し、どの主体がどのように運用コストを負担するのも問題となるが、配信ポリシーに価格機構を用いた市場機構による資源割当てを応用することにより、コストの問題を系に含めて最適化問題として記述することができ、また、流通するコンテンツの価値を考えた応用も可能であると考えられる。

以下、市場機構分散資源割当てに基づいて、エンドユーザの効用関数の記述、および、コンテンツ決定アルゴリズムについて記述する。

4.1 市場モデル

消費者エージェント・生産者エージェントと消費者エージェントの支払う対価、生産者エージェントの生産するキャッシュサービスの関係を図 3 に示す。

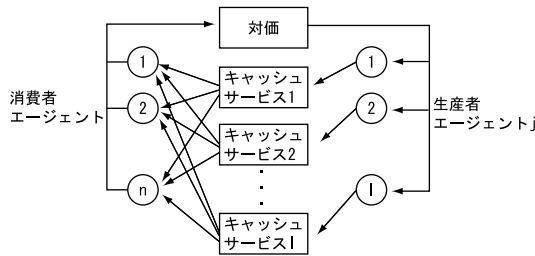


図 3 CDN における市場モデル

図 3 において、左側の円は消費者エージェントを示し、右側の円は生産者エージェントを示す。また、図中の長方形は各エージェント間で流通する財を表す。

流通するコンテンツは、直接提供されるのではなく、ネットワークやストレージなどを含めて、コンテンツキャッシュサービスとしてエンドユーザに提供される。消費者エージェントは、対応するエンドユーザの選好情報をエンドユーザから取得し、キャッシュサービスに対する需要を決定する。生産者エージェン

トは、複数あるキャッシュのうちの 1 つであり、このキャッシュが、エンドユーザの対価を用いてストレージやネットワークなどキャッシュの各資源からコンテンツキャッシュサービスを生成する。

以下、エンドユーザを代表する消費者エージェントと、キャッシュを代表する生産者エージェントの行動について記述する。

4.2 消費者エージェント

エンドユーザの効用関数の例として、CES 型の効用関数をあげる。ユーザ数を n とし、エンドユーザ k ($1 \leq k \leq n$) に対応して消費者エージェント k がエンドユーザの効用情報を持つとする。消費者エージェント k の CES 効用関数 u_k は以下のように表される。

$$u_k(x_{1k}, \dots, x_{lk}, e_k) = \left(\sum_{i=1}^l a_{ik} x_{ik}^{\frac{\rho}{\rho-1}} + a_{ek} e_k^{\frac{\rho}{\rho-1}} \right)^{\frac{\rho-1}{\rho}} \quad (1)$$

式 (1) において、 x_{ik} は、消費者エージェント k に割り当てられる資源 i の量、 l は資源の種類の数、 e_k はキャッシュサービス利用のために用いる対価としての初期財を表す。

すなわち、ネットワーク上には映画や音楽などの l 種のコンテンツがあり、その各コンテンツの中で、エンドユーザが x_{ik} だけの個別のコンテンツを要求すると理解できる。 a_{ik} は資源 i が消費者エージェント k にとってどの程度価値があるのかを示し、 ρ はある一つの資源が他の資源をどの程度代替するかを示す。

消費者エージェントは、消費者エージェントが初期に持つ予算を制約条件としてこのような効用関数を最大化するように行動する。すなわち、 p_i を資源 i に付けられた価格^{†2}とすると、

$$\begin{aligned} \max u_k(x_{1k}, \dots, x_{lk}, e_k) \\ \text{s.t. } \sum_{i=1}^l p_i x_{ik} \leq e_k \end{aligned} \quad (2)$$

の最大化問題の解としてエンドユーザの選好を得ることができる。

式 (2) のためのエンドユーザの効用情報の取得は、キャッシュコンテンツ決定のプロセスを行う前に、ブラウザやマルチメディアプレイヤ上のカテゴリ選択などで行うことが可能であると考えている。

^{†2} 提供されるキャッシュサービスの価格にあたる

4.3 生産者エージェント

エンドユーザの選好を効用関数で表わし、前述のプロトコルによってキャッシュに集められると、キャッシュはこれらの効用関数の総和として表記されるような社会的効用を最大化する。

キャッシュを代表する生産者エージェントは、生産関数であらわされる技術内でキャッシュの資源からコンテンツキャッシュを生産する。各コンテンツ i に対しては、 $\sum_{k=1}^n x_{ik}$ だけの需要があり、 $\sum_{k=1}^n p_i x_{ik}$ だけの財が割当てられる。キャッシュは、 $\sum_{i=1}^l p_i \sum_{k=1}^n x_{ik}$ を用いて、キャッシュコンテンツとストレージを購入し、キャッシュサービスを構築する。キャッシュサービスは、各コンテンツ種別 i 毎に $\sum_{k=1}^n x_{ik}$ だけの需要がなされ、この範囲内で、個別のコンテンツを購入し、キャッシュに置く。

各コンテンツ i に対するキャッシュ j の行動は、コンテンツ価格を p_i^c 、ストレージ価格を p^s 、コンテンツの購入量、ストレージの購入量、そこから生産されるキャッシュサービスをそれぞれ、 c_{ij} 、 s_{ij} 、 y_{ij} として、式 (3) の最大化問題の解として与えられる。

$$\max \sum_{i=1}^l (p_i y_{ij} - (p_i^c c_{ij} + p^s s_{ij})) \quad (3)$$

$$y_{ij} = f_j(c_{ij}, s_{ij})$$

ここで、 $f(c_{ij}, s_{ij})$ は、コンテンツとそれを提供するためのストレージ等の資源からキャッシュサービスを生産する生産関数となっている。

式 (3) の均衡解を y_{ij}^* とする。 y_{ij}^* を与える 2 財 c_{ij} 、 s_{ij} の組みを (c_{ij}^*, s_{ij}^*) で表わすと、 c_{ij}^* 、 s_{ij}^* は、式 (4) を満たす。

$$\begin{cases} \frac{p^s}{p_i} = \frac{\partial}{\partial s_{ij}} f(c_{ij}^*, s_{ij}^*) \\ \frac{p_i^c}{p_i} = \frac{\partial}{\partial c_{ij}} f(c_{ij}^*, s_{ij}^*) \end{cases} \quad (4)$$

式 (4) を満たす解 (c_{ij}^*, s_{ij}^*) が得られても、キャッシュは必ずしもサービスを提供するとは限らない。価格の調整が終了し、最終的な均衡 (c_{ij}^*, s_{ij}^*) が得られた時、式 (5) を満たしている必要がある。

$$\sum_{i=1}^l (p_i^c c_{ij} + p^s s_{ij}) \leq \sum_{i=1}^l p_i \sum_{k=1}^n x_{ik} \quad (5)$$

式 (5) は、キャッシュがキャッシュサービスを提供

することにより、利益が得られることを示しており、この式が満たされない場合、キャッシュサービスのみを独立して運用することは、損失が生じることから不可能であると言える。

また、ストレージやネットワーク等のリソースにおいて大容量なものを用意することが困難であるなどの収穫逨減の性質を示すのに対し、コンテンツは従量課金あるいは固定の料金体系となりうるため、生産関数が不連続、あるいは、収穫不変となりうる。

図 4 に、 x 、 y 軸をキャッシュサービスを生産するために必要なストレージなどのリソース、及び、コンテンツをとり、 z 軸に生産されるキャッシュサービスをとり、このような狭義凸関数とならない生産関数を例示する。

図 4(a) は、コンテンツの費用が定数で、一定額支払えば制限なくコンテンツを利用できる場合であり、図 4(b) は、コンテンツの費用は従量制で、利用するコンテンツ量に対して線型にコストが増加する場合を示している。このようなコンテンツに特有の性質により、式 (4) より得られる均衡解は、大域的な最適解ではなく局所的な最適解に留まる、あるいは、利潤に上限がなくなり式 (3) を満たす解が存在しない可能性がある。

以上のように、生産者エージェントは、

1. キャッシュサービスを提供して正の利潤を得るための条件
 2. 最適なコンテンツ集合が得られるための条件
- を考えて実装する必要がある、今後この条件を明らかにしていく予定である。

4.4 価格調整機構

各生産者エージェント・消費者エージェントは対応する財について最大化問題を解き、その解を競売人エージェントに入札する。競売人エージェントは、生産者エージェントと同じくキャッシュシステム内のソフトウェアエージェントであり、各資源・コンテンツの需給がつかうように価格を調整する。

競売人エージェントは、 $\sum_{k=1}^n x_{ik}$ と、あるキャッシュ j での y_{ij} が表明された場合に、関数 θ_i を単調増加関数として、新しい価格を式 (6) に従い変更し、

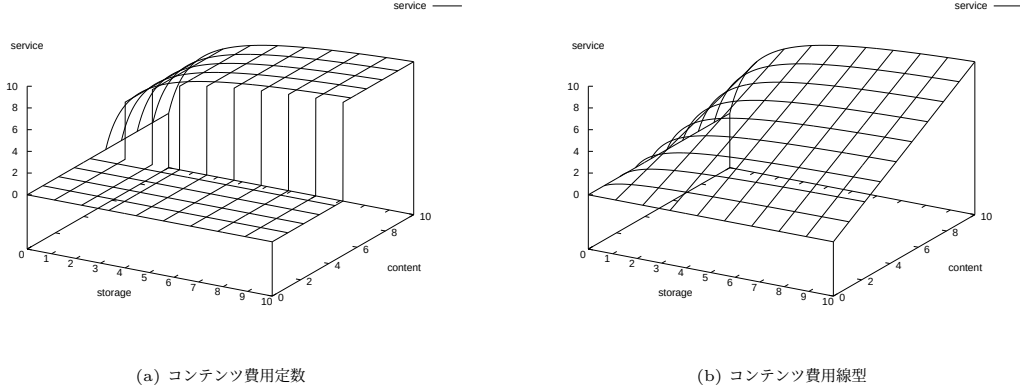


図 4 収穫逓減でない生産関数

キャッシュ j において $\sum_{k=1}^n x_{ik} - y_{ij} = 0$ となる均衡解 $\mathbf{p}^* = (p_1^*, \dots, p_l^*)$ を探す. この時の y_{ij} が, コンテンツ種別 i に対するキャッシュサービスの最適供給量となる.

$$\forall i, \frac{dp_i}{dt} = \theta_i \left(\sum_{k=1}^n x_{ik} - y_{ij} \right) \quad (6)$$

競売人エージェントは決定された新しい価格を各消費者エージェント・生産者エージェントに通知する.

4.5 計算手順

消費者エージェント・生産者エージェントはその時点の価格における最大化問題の解を計算し, 競売人エージェントにその解を伝える. 競売人エージェントは, キャッシュサービスの最適供給量を得られる均衡価格を計算し, 消費者エージェント・生産者エージェントに伝える. この過程を収束するまで繰り返し, 得られたキャッシュ集合に対応するように各コンテンツをキャッシュする. 図 5 にこのアルゴリズムを示す.

市場が適切な性質を持つ場合, 上述のプロセスは収束し, 需要・供給量は均衡することが知られている. また, 均衡点での需要・供給量は実際に割り当てが可能である, すなわち, 実行可能な解であり, パレート最適性を示すことが知られている.

パレート最適性は, あるエージェントがそれより有利な資源配分を得る時は, 必ず他のエージェントが不利になる性質であり, パレート最適な割り当ては無駄

の無い割り当てがなされていることを表している.

5 おわりに

本稿では, コンテンツデリバリネットワークにおけるコンテンツのキャッシュポリシーについて記述した.

マルチメディアコンテンツが容易に作成され公開されるようになり, サーバの負荷, また, ネットワークトラフィックが増大している現在, コンテンツデリバリネットワークはこの問題に対して有効な解決策と言える. より多くのコンテンツに対してコンテンツデリバリネットワークを適用しエンドユーザにより速いサービスを提供するためには, どのコンテンツをキャッシュするのかの指針を定めることが必要となると考えられる.

本稿で取り上げた手法により, ユーザの要求に適応し, 経済機構を背景としたコストモデルを持つキャッシュの指針が与えられ, コンテンツデリバリネットワークに貢献するものであると考えている. 今後,

1. 式 (3) において, 収穫逓減でない性質の市場における最適性の検討
2. 本稿のモデルに対応したシミュレーションによる評価
3. ネットワーク事業者, コンテンツ配信者, 及び, エンドユーザの各主体へのコスト負担を含めて, ビジネスモデルを考慮したモデルへの改善

の 3 点を行い, モデルの有効性を確認していく予定である.

```

begin Auctioneer Agent
  begin loop
    for each Producer Agent, Consumer Agent
      send Price Vector
    end of for
    for each Producer Agent
      receive Production matrix
    end of for
    for each Consumer Agent
      receive Demand matrix
    end of for
    if balance ( Production matrix, Demand matrix) exit
    Price Vector =  $\theta$ ( Price Vector, Demand matrix, Production matrix)
  end of loop
end of Auctioneer Agent

begin Producer Agent
  begin loop
    receive Price Vector from Auctioneer Agent
    Production matrix = Calculate production ( Price Vector )
    send Production matrix to Auctioneer Agent
  end of loop
end of Producer Agent

begin Consumer Agent
  begin loop
    receive Price Vector from Auctioneer Agent
    Demand matrix = Calculate demand ( Price Vector )
    send Demand matrix to Auctioneer Agent
  end of loop
end of Consumer Agent

```

図 5 各エージェントのアルゴリズム

謝辞

本研究の機会を与えて下さり、また、的確な御指導を賜りました京都大学大学院情報学研究科社会情報学専攻の石田 亨 教授、NEC USA の 原 良憲 様、京都大学大学院情報学研究科社会情報学専攻の亀田 弘行 教授に深く感謝申し上げます。

参考文献

- [1] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S.: Web Caching and Zipf-like Distribu-

tions: Evidence and Implications, 1998.

- [2] Karger, D. R., Lehman, E., Leighton, F. T., Panigrahy, R., Levine, M. S., and Lewin, D.: Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web, *ACM Symposium on Theory of Computing*, 1997, pp. 654-663.
- [3] Wellman, M.: A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems, *Journal of Artificial Intelligence Research*, Vol. 1(1993), pp. 1-23.
- [4] Yamaki, H., Wellman, M. P., and Ishida, T.: A market-based approach to allocating QoS for multimedia applications, *Proceedings of the First International Conference on Multi-Agent Systems*(Lesser, V.(ed.)), MIT Press, 1995.

口コミによる分散型情報収集システム

WAVE を起こそう—Word-of-mouth-Assisting Virtual Environment

吉田 匡志 伊藤 雄介 沼尾 正行

Abstract

情報処理装置として出発したコンピューターは、近年、コミュニケーションメディアとしての役割を果たすようになってきた。しかし、コミュニケーション上のトラブルなどやコミュニティ参加への心理的不安が活発な情報交換を妨げる原因となっている。また、氾濫した情報の中から有用な情報を獲得するためには大変な労力を必要とする。そこで、本稿では、口コミを電子コミュニティ上に再現することで効率の良い情報収集や円滑なコミュニケーションを支援するシステム、WAVE (Word-of-mouth-Assisting Virtual Environment) を提案する。また、このシステムを実装し実験した結果、いくつか興味深い現象を観察した。

Recently, computers play an important role not only in knowledge processing but also as communication media. However, they often cause troubles in communication, since it is hard for us to select only useful pieces of information. To overcome this difficulty, this paper proposes a new system, WAVE (Word-of-mouth-Assisting Virtual Environment), that helps us to communicate and to collect information by word of mouth, and shows some interested phenomenon in the experiment.

1 はじめに

現在、WWW や電子メールといった電子メディアによるコミュニケーションが大変賑わいを見せている。情報処理装置として出発したコンピューターは、通信技術やヒューマンインターフェースなどの発達により、コミュニケーションメディアとしての役割を果たすようになってきた。

しかしその一方で、フレーミング^{†1}などのコミュニケーション上のトラブルや、コミュニティ参加への心理的不安といったものが実世界と比べて顕著に表れ、活発な情報交換を妨げる原因となっており、コンピューターを介したコミュニケーションと実世界におけるコミュニケーションがどう違うのかといった議論 [1][2] が盛んに行われている [3]。

また、計算機ネットワーク上には様々な情報が氾濫し、ユーザーにとって有用な情報を獲得するのに大変な労力を必要とする。情報を発見するための最も身近なツールとして検索エンジンがあるが、最もカバー率の大きいとされる Alta Vista でもせいぜい 4 割から 6 割程度であると言われており、広大な WWW 空間に氾濫した情報をすべて把握するのは非常に困難である。それにどのような情報が欲しいかユーザー自身が明確に分からない場合も多い。

そこで、情報の内容を解析しユーザーが文章中のどの部分に対して興味を持っているか推定することで同様の部分を持つ情報を有用な情報として提供する

東京工業大学 大学院情報理工学研究科 計算工学専攻
{myoshida, blanke, numao}@nm.cs.titech.ac.jp

^{†1} コミュニケーションの中で現れる、ののしり、侮辱、強調された表現

Content based filtering [4][5] や、評価傾向の似ている他ユーザーの持つ情報を参考にして有用な情報を提供する Colaborative filtering [6][7] などの情報フィルタリングの研究が盛んに行われてきている。しかし、これらは数値的な計算だけで人の心の動きや人間関係を解釈しようとする傾向が強く、機械的な精度を上げて限界がある。

これに対して、大量の意見を要約して質の高い情報を提供するマスメディアと自由に情報発信が可能でインタラクティブ性を持つ電子メディアを融合した新しいメディア [8] や、コミュニティの可視化 [9][10][11] など、人の心の動きや人間関係を考慮しコミュニティの形成を支援する研究もある。コミュニティの形成を支援することは情報収集の支援にもつながる。しかし、情報フィルタリングのようにユーザーにとって有用な情報を収集するといったことまでは考慮されていない。

そこで、本稿では、口コミを電子コミュニティ上に再現することで、効率のよい情報収集や円滑なコミュニケーションを支援するシステムを提案する。

2 口コミの特徴

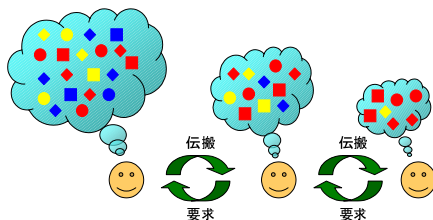


図1 口コミのイメージ

口コミは、不特定多数の人々に均一な情報を伝えるマスメディアと異なり、個人間で双方向的に情報の伝達が行われ [12]、図1のように、氾濫する情報の中から多くの人々の評価と伝播を経ることで有用な情報だけが人々の間に広まっていく。口コミによる情報伝播では、相手の興味のありそうな情報であるかを判断して伝えることや、個人の情報収集では見逃してしまうような重要な情報を伝えることができる。さらに、情報を受け取る側は情報提供者に対して、信頼できる人か

や、その話題に関して専門性を持っている人であるかといった評価をあらかじめ持っている。また、本人の経験を元に重要な部分が強調されるので、情報の質が高くなる。よって容易に情報の価値判断ができる。

しかしその反面、「平均化」（伝達要素の減少）や「強調化」（少数要素の強調）、「同化」（予期的枠組みへ一貫する方向への内容変化）などの情報内容の変容が起こりやすく、うわさや流言など発生しやすい。また、人間関係のつながりにより情報が伝播していくので、強いインパクトのある流言でもない限りは狭い範囲に限定されがちである。

口コミによる情報伝達については、いろいろな研究がされており [13]、その一つに社会ネットワーク分析^{†2} [14] の研究分野がある。ここでは、親友など頻繁に対面接触する緊密な人間関係を「強い紐帯（ちゅうたい）」と呼び、まれにしか対面接触をしない薄い人間関係を「弱い紐帯」と呼ぶ。また、広がる人間関係の網の中で、派閥のように互いに直接結びつきあっている人間関係（強い紐帯）の集合を「クリーク」と言う。そして、クリークの間を結びつける弱い人間関係（弱い紐帯）が「ブリッジ」である。

一般に、口コミはクリークの中で活発に行われ、人間関係の紐帯が強いほうが影響力があり、信頼性が高く有効である。また、専門性を認知されたり信頼性が高い方がより説得的であるなどの結果も出ている [15]。しかし、転職時に弱い紐帯を通じて情報を得た人の方が転職後の満足度が高く、弱い紐帯は強い紐帯よりも有効であったという興味深い結果も報告されている [16]。クリーク内では情報伝達が早いですが、同じような興味が持った人々が集まっているので似たような話題についての情報交換がされやすい。新しい情報は、ブリッジを通じてクリークへと導入される。

これらの口コミの諸性質は、電子コミュニティ上においても当てはまると考えられる。それと同時に、電子メディア特有の性質が実世界の口コミが持っていた

^{†2} 集団成員間のコミュニケーションの構造を見出す分析手法。誰と誰がどれくらいコミュニケーションをとったのか、データを採り、これをグラフ化し、グラフ理論を用いて、中心性、密度といった指標を計算することによって、クリークやブリッジといった構造的役割を発見する。

欠点を補うと考えられる。電子メディアにおける情報の伝播はオリジナルのコピーの転送や URL のリンク情報の伝達といった形で行われる。このため実世界における情報伝播よりも正確な情報伝播が行われ、情報の変容が起こりにくい[17]。また、口コミは狭い範囲に限定されがちであるが、電話により口コミの伝達速度が急激に上昇し、その伝達範囲を大きく広げた[13]ことから、電子コミュニティではさらに伝達速度が上昇し、その伝達範囲も広がる。

3 WAVE のしくみ

3.1 口コミ成立の要件

本稿で提案するシステム、WAVE^{†3}(Word-of-mouth-Assisting Virtual Environment) は、電子コミュニティ上において口コミを再現することで、効率の良い情報収集や円滑なコミュニケーション支援を行う。過去に、口コミの考え方を導入し検索エンジンの分散管理を行った研究[18]があるが、口コミ成立の要件や実現すべき目標が本稿と異なる。なお本稿で意図する口コミとは以下のようなものである。

- 情報のプッシュとプル統合
- 信頼関係にある人同士での情報交換が可能
- 評価情報を伴う情報伝達

この要件を満たすことで、2 章で説明した性質を持って情報が伝播していく。また、口コミの成立を支援するためにユーザーインターフェースにも工夫を施した。以下では WAVE のしくみとその特徴について図 2 に沿って説明していく。

3.2 情報の発信

各ユーザーは、自分が知っている新しい情報を発信することができる。図 3 のように、自然言語による情報だけでなく、Web ページや画像データの URL 情報を付加することができたり、情報を閲覧するユーザーが情報の内容を判断しやすいように、情報を簡単に分類するためのカテゴリを割り当てるようになって

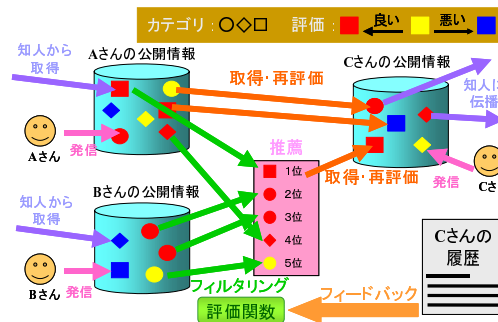


図 2 WAVE における口コミの再現

いる。また、情報に対して 1～5 (1 が一番悪く、5 が一番良い) の評価値を与える。発信した情報は、自分の情報として他ユーザーに公開され、WWW やメーリングリストのように、自分なりの情報を多数の人々に向けて自由に発信することが可能である。

3.3 情報の公開

図 2 のように、自分が発信した情報や他ユーザーから取得した情報は、自分の情報として蓄積される。また、自分の情報は他ユーザーに公開され、隣接するユーザーへ自動的に情報が流れていく。例えば、図 2 の C さんが知人の A さんや B さんの持つ WAVE 専用のアドレス (”ユーザー名@ホスト名:ポート番号”の形) を図 4 の画面で”お友達リスト”としてシステムに登録すれば、C さんは A さんや B さんに隣接するユーザーとなって、A さんや B さんから自動的に流れてきた情報を簡単に閲覧することができる。WAVE では、情報をもらいたいユーザーをお友達リストに登録したり、逆に、自分の情報を伝えたい人に自分のアドレスをお友達リストに追加してもらうように通知したりすることができる。

これにより、情報のプルとプッシュを統合することができる。また、ユーザー同士の人間関係が強く現れ、相手の専門分野や信頼性を判断することができるので、WWW などに欠落していた情報源の信頼性を高めることができる。さらに、自分の信頼する隣接ユーザーとのやり取りがほとんどなので、安心してコミュニティに参加できる。

^{†3} WAVE には、口コミが波のように伝播していくというイメージと、このシステムが、WAVE を起こし、世界中の人々に使ってもらえるようなコミュニケーションメディアとなつてほしいとの願いがこめられている。

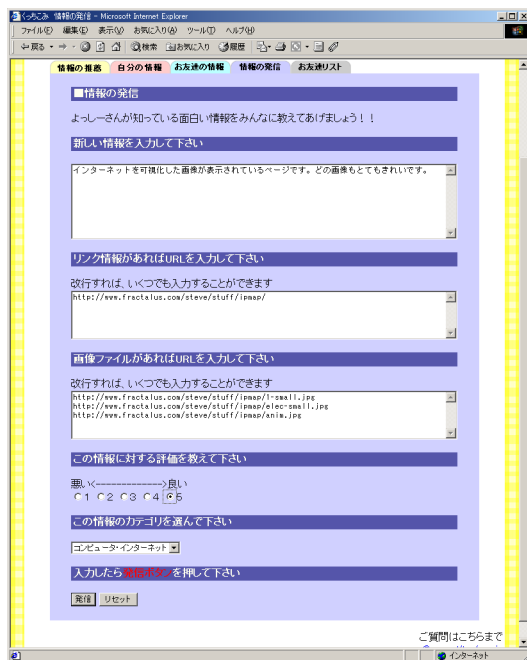


図 3 情報の発信画面

3.4 情報の評価と取得

図 2 で、C さんが A さんから流れてきた情報の中から興味を持った情報があれば、図 5 の画面のようにその情報に関する詳しいデータを見ることができる。このとき、その情報に新しい評価をつけたり、自然言語による付加情報としてコメントを与えることができる。この画面では、自然言語による情報だけでなく画像やハイパーリンク、情報提供者による評価・コメントなどが表示されるのでその情報が有用かどうかの判断を支援するようになっている。C さんが新しい評価やコメントを与えると、A さんの情報は自動的に取得され C さん自身の情報として公開される。そして、それがまた他のユーザーに伝播していく。

このように WAVE では情報の公開、閲覧、評価、取得をシームレスに行うことができるので、情報のアップロードや情報の存在を人々に知らせるため手間を軽減し、情報のプッシュを促進する。また、BBS やメーリングリストにおける ROM(Read Only Member) が持つような積極的な参加への心理的抵抗感も軽減する。このことは、情報伝播において重要な役割を果たすブリッジの存在を維持することにつながる。



図 4 お友達リスト画面

多くの人間を経由して情報の評価と取得が繰り返し行われるうちに段階的にフィルタリングが行われ、有用な情報だけがネットワーク上に広まる。相手を侮辱するような情報やチェーンメールのような情報を発信しても周りから取得されず自分の情報としてとどまり広がりにくい。また、同じような嗜好を持つ人間はクリークとして集まりやすいので情報の流れは指向性を持つ。そしてブリッジを介して情報がひとたび自分の属するクリークに到達すれば、自分もその情報を得られる。WAVE では、人間のコミュニティそのものが分散化された情報収集システムとして機能し、グローバルな情報交換ネットワークを形成されるといえる。

3.5 情報の推薦

”お友達リスト”に登録したユーザーの数が増加したり、ひとりのユーザーが公開する情報の数が増加したりすることによって、流れてくる情報をすべて閲覧するのは負担となってくることが予想される。

そこで、図 2 のようにユーザーの閲覧履歴等をもとに動的に作成した評価関数により、公開されている情報の中から有用であると思われる情報の一覧 (図 6)



図 5 情報の詳細画面



図 6 情報の推薦画面

を提示し情報の閲覧にかかる負担を軽減する。なお、評価関数は以下の項目を基準にして作成した。以下の項目の値が大きくなると、評価関数の値が大きくなり、ユーザーにとって有用な情報として推薦される。

- 情報提供者が与えた情報に対する評価
口コミによって流れる情報は、情報提供者によってすでに 1~5 (1 が一番悪く、5 が一番良い) の評価値が与えられている。情報提供者の評価値が高ければ、他のユーザーにとっても評価の高い情報であると考えられる。
- 情報提供者に対するユーザーの評価
ある人の情報を見たり取得する回数が多い時、その人に対して高い評価をしているので、その人から流れてきた情報の評価も高いと考えられる。このとき、コンピューターに詳しい人からはコンピューターに関する情報を取得しがちであるというように、ユーザーが取得する情報の分野は情報提供者によって偏りがあり、提供者の評価には提供者の持つ専門性も考慮する必要がある。そこで、すべての情報提供者の中で、今までに、その情報提供者の情報をどれくらい閲覧・取得

したかや特に今回の情報と同じカテゴリの情報はどれくらい閲覧・取得したかクリック率を計算し、この値が大きいほど情報提供者の評価が高いとする。さらに時間が経つと、情報をいつも参照する人が変わることがあるので、情報提供者の情報を最後に見た日数が経過するにしたがって、その情報提供者の評価が低くなるようにする。

- ユーザーの嗜好
ユーザーが興味のあるカテゴリに関するものほど、情報に対する評価は高いと考えられる。そこで、今回の情報と同じカテゴリの情報に対して今まで与えてきた評価値の平均や、すべてのカテゴリの中で今回の情報と同じカテゴリの情報をどれくらい閲覧・取得しているかクリック率を計算し、これらの値が大きいほど情報の評価が高くなるようにする。
- 情報の鮮度
新しい情報ほどユーザーの評価は高いと考えられる。情報が取得・公開されてからの日数が経ったものほど情報の評価は下がるようにする。
- 情報が伝達した人数

多くの人の間を伝達した情報ほどユーザーの評価は高いと考えられる。伝達した人数が多いほど情報の評価が高くなるようにする。

WAVE における情報の推薦は、Collaborative filtering のように評価傾向の似ている他ユーザーの持つ情報を参考にして有用な情報を提供するのではなく、隣接ユーザー間で情報をプッシュする側とプルする側のマッチングを行う。現段階では本システムは、この情報はあの人に教えたいというように、情報ごとに特定のユーザーを指定してプッシュすることができないので、情報の推薦がこの機能を補う。

3.6 システムの分散化

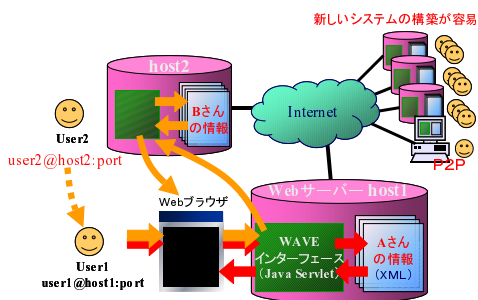


図7 システムの分散化

図7のように、WAVE ではシステム自身の分散化も行っている。システムは Java サーブレットを用いて実装されており、Web サーバー上で動作する。ユーザーは、Web ブラウザを介してユーザー登録をしたホストにアクセスするだけで、情報のやり取りをすることができる。

自分の情報は、ユーザー登録したホスト上に蓄積される。なお、この情報は、次世代の Web の標準言語となる XML により保存されている。これにより、既存の Web ページからの情報を有効利用することなどが将来的に可能となり、WAVE は WWW を包括するメディアとなり得る。

分散化されたシステム同士での連携の仕組みは非常に単純である。図7を例に説明すると、User1 は、host1 で動作するシステムにユーザー登録しているので、host1 へアクセスすることで、情報のやりとり

を行うことができる。また、他のホスト host2 を利用している友人の User2 の情報が閲覧したいときは、登録した User2 の WAVE 専用のアドレスからそのユーザーが持つ情報の存在位置が一意に定まるので、User2 の情報が host1 に存在するかのように情報の存在位置を気にすることなく取り扱うことができる。

分散化によりシステムをどこでも自由に構築できるので、ネットワークの負荷分散や、CPU、ストレージなど計算資源の有効活用が可能である。また、ユーザー数が増加しても新しくシステムを構築すればよく、スケーラビリティが高い。したがって、ユーザー数の増加が促進され、より効率的な情報収集を WAVE で行うことが可能になる。さらに、WAVE を Peer-to-Peer 方式^{†4}で実装すれば、一般的になりつつある個人のインターネットへの常時接続環境において広く使われると考えられる。

4 実験

WAVE を実装し、被験者 33 名で 20 日間運用した。その後、実験データを解析し、社会ネットワーク分析のためのグラフ可視化ツール KrackPlot [19] を用いて、図8のように、WAVE 上でのコミュニティにおけるグラフを作成した。

ノードはユーザーを表しており、情報を最初に発信した回数によって色分けされている。この場合、myoshida や blankey、roy、t-sugie などがオピニオンリーダーとなってコミュニティに話題を提供している。ノード間を結ぶアークは、あるユーザーから別のユーザーへ矢印の方向に情報が取得されたことを表している。線の太さは、情報を取得した回数である。ノードの入次数が多い場合、ユーザーは多くのユーザーを信頼し情報を獲得していると言える。逆にノードの出次数が多い場合は、多くのユーザーから信頼されるような情報を提供していると言える。

図の中心にあるノードからアークが放射上に広がっ

^{†4} クライアント・サーバー型のネットワークと違い、ネットワーク上のコンピュータのそれぞれが、サーバーでありクライアントとなる。よって、集中的に処理を行なうサーバーを設置することなく、各ネットワーククライアントが持つ資源をお互いに共有する事が可能である。

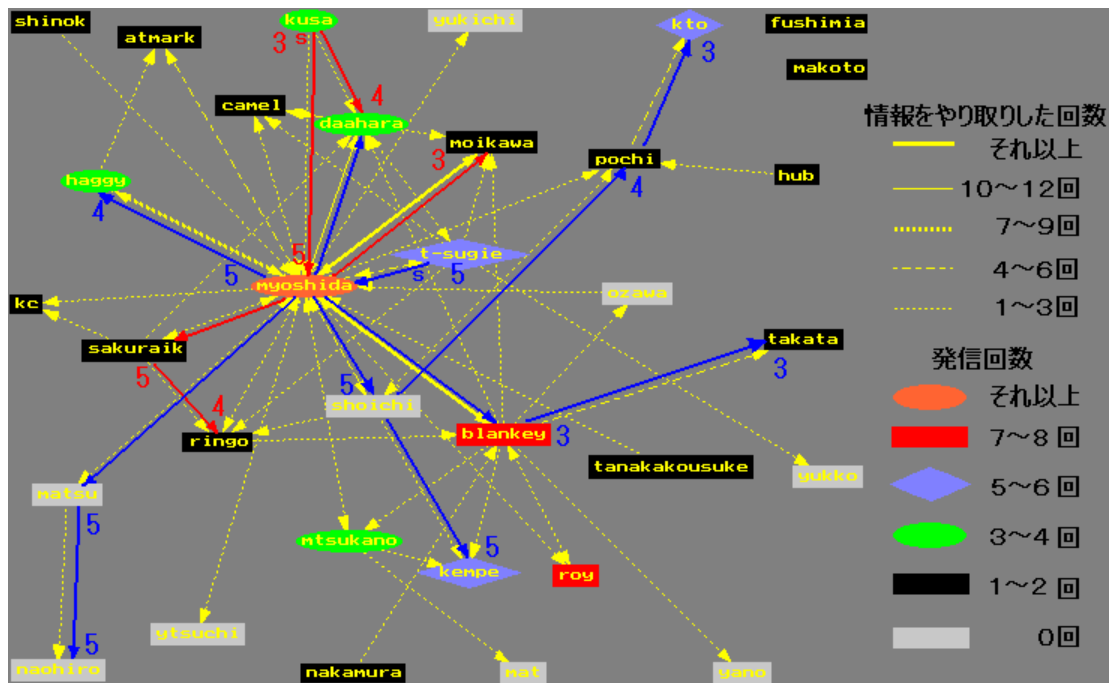


図8 WAVE 上でのコミュニティ

しており、中心にいる myoshida や blankey の周辺で活発な情報のやりとりが行われていることが分かる。また、アークによって結ばれた三角形がいくつも確認できる。このような三者関係 (triad) [14] は小さなクリークで結びつきが強く、実世界におけるコミュニティを照らし合わせると、この三角形がいくつも結びついて大きなクリークを形成しているのが分かった。

さらに、情報が伝播していった例を2つ、それぞれ赤と青の線でトレースした。ノードの脇にかかれている数字は、ユーザーがその情報に対して与えた評価値 (1~5) である。ここで、情報が伝播するにしたがって徐々に評価が下がっていることが分かる。また、青い線の情報は、matsu、naohiro の辺りでは評価が高いが、blankey、takata や poti、kto では低くなっている。これは、集団内では似たような価値観を持った人が多いことや自分の評価を決定するときに情報提供者の評価を参考にしていないからではないかと考えられる。情報が伝播されるうちに評価が減少していくと、取得される価値がないと判断され伝播が途絶える。一方、その情報に興味をもった人たちの間には

広がっていく。

また、情報は人間関係に沿って流れ、myoshida、shoichi のように情報の経路が枝分かれしているところで、複数の集団 (クリーク) へ情報が伝播し、集団内では急速にその情報が広がるといった現象が起こっている。この場合、図から弱いつながりとしてブリッジの存在を確認するのは難しいが、そのようなユーザーは実世界において、サークルや研究室など異なる集団を結びつける人物であることが分かった。このような人物がゲートキーパー (gatekeeper) [20] として働いていると考えられる。ユーザーに使用してもらった感想を聞くと、システム上で流れてきたある情報について、同じ集団に属するユーザーの間で実世界において初めて話題になった時にすでに全員が知っていたなど、その現象をユーザー自身で実感することができたようだ。このように、2章で説明した口コミの性質が観察され、WAVE の有効性を確認することができた。

5 まとめ

本稿では、口コミを電子コミュニティ上に再現することで、効率のよい情報収集や円滑なコミュニケーションを支援するシステムの提案を行った。また、実験データから作成した WAVE 上でのコミュニティのグラフから WAVE の有効性を確認することができた。

今後も引き続きシステムを運用し実験を行う。

現在、<http://www.nm.cs.titech.ac.jp:12581/wom/>で、システムを公開している。興味を持たれた方は実験に御協力頂きたい。また、本システムを配布し、複数のホストで動作させ、分散環境を実現したい。

その後、本稿で行ったようなコミュニティの可視化を行うことで、クリーク、ブリッジなどの構造的役割や情報の伝播の仕方などさらに詳しい分析を行ったり、ユーザーにアンケートを行うことでシステムの使用感について調査し、システムを評価する。情報の推薦については、推薦により情報を取得した割合や推薦により取得した情報の順位の変化、ユーザーへのアンケートなどによりその妥当性について評価する。

また、システムの改良についても行っていきたい。現状では、カテゴリは情報の発信時のみ付与するようになっているが、取得時にも各ユーザー独自のカテゴリを付与する。カテゴリをフォルダとみなし階層構造を許すようにすると、自分の持っている情報をフォルダごとに自由に整理することも可能となる。この際、情報の推薦については、階層構造内で何親等になるかという距離を定義することでカテゴリ間の類似度が計算できるので、それを他のカテゴリ内の情報への評価として活用すれば、カテゴリに分類したことによるサンプル数の減少を補うことができる [7]。

参考文献

- [1] Sproull, L. and Kiesler, S.: コネクションズ: 電子ネットワークで変わる社会, アスキー出版, 1993.
- [2] Lea, M.: Contexts of Computer-Mediated Communication, Harvester Wheatsheaf, pp.30-65, 1992.
- [3] 野島 久雄: 電子メディア社会の心理学, 情報処理学会論文誌, Vol.40, No.1, pp.66-70, 1999.
- [4] 溝口文雄, 大和田勇人. 帰納学習に基づく情報フィルタリング. 人工知能学会全国大会 (第10回) 論文集, 1996.
- [5] Pattie Maes: Agents that reduce work and information, CACM, Vol.37, No.7, pp.30-40, 1994.
- [6] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. CSCW '94 Proceedings. pp.175-186, 1994.
- [7] 沼尾 正行, 横山 甲: 階層化された知識の継承による情報フィルタリング, 情報処理学会「知能と複雑系」研究会, 1999.
- [8] 西田 豊明, 畦地 真太郎, 藤原 伸彦, 角 薫, 福原 知宏, 矢野 博之, 平田 高志, 久保田 秀和: パブリック・オビニオン・チャンネル, 第2回 CMCC 研究会シンポジウム, http://www.brl.ntt.co.jp/people/nojima/CmCC/Resources/CmCC_Symposium_Papers/cmcc0206-nishida.pdf, 1999.
- [9] 藤田 邦彦, 亀井 剛次, Eva Jettmar, 吉田 仙, 桑原 和宏: ネットワークコミュニティの可能性-Community Organizer 評価実験結果報告-, 第3回 CMCC 研究会シンポジウム, http://www.brl.ntt.co.jp/people/nojima/CmCC/Resources/CmCC_Symposium_Papers/cmcc03-fujita.pdf, 2000.
- [10] 高橋 正道, 北山 聡, 金子 郁容: ネットワーク・コミュニティにおける組織アウェアネスの計量と可視化, 情報処理学会論文誌, Vol.40, No.11, p.3988-3999, 1999.
- [11] 館村 純一: 協調型情報探索を支援する仮想評者とその視覚化, WISS'99, 1999.
- [12] 岡堂 哲雄: 現代のエスプリ別冊 社会心理用語事典 p.91, 至文堂.
- [13] 中村 功: 現代のエスプリ別冊 特集「流行... ファッション」流行と口コミ電話, pp.199-209, <http://cc.matsuyama-u.ac.jp/~nakamura/espri.htm>, 2000.
- [14] 安田 雪: ネットワーク分析, 新曜社, 1999.
- [15] Bristol, J.: Enhanced explanations of word of mouth communications; the power of relations, Research in Consumer Behavior, Vol.4, pp.51-83, 1990.
- [16] Granobetter, M.: Getting A Job, 1974 (渡辺深訳「転職」ミネルバ書房, 1998).
- [17] 柴内 康文: 電子メディア社会における情報伝播, 第2回 CMCC 研究会シンポジウム, http://www.brl.ntt.co.jp/people/nojima/CmCC/Resources/CmCC_Symposium_Papers/cmcc0205-shibanai.pdf, 1999.
- [18] 大谷 武, 南 俊朗: 口コミによる情報資源探索, MACC'1997 一般セッション (情報検索支援), <http://www.kecl.ntt.co.jp/csl/msrg/events/macc97/ohtani.html>, 1997.
- [19] KrackPlot home page, <http://www.contrib.andrew.cmu.edu/~krack/>.
- [20] 後藤 滋樹, 野島 久雄: 人間社会の情報流通における三段構造の分析, 人工知能学会誌, Vol.8, No.3, pp.348-356, 1993.
- [21] 伊藤 雄介, 吉田 匡志, 沼尾 正行: 口コミ支援システムの実験, 情報処理学会「知能と複雑系」研究発表会, 2001.
- [22] 伊藤 雄介, 吉田 匡志, 沼尾 正行: 多くの人の評価を経て情報が吟味される口コミ支援システム, 第15回人工知能学会全国大会, 2001.

知識の階層構造を用いた協調的情報推薦ネットワークの提案

濱崎 雅弘 武田 英明 河野 恭之 木戸出 正継

1 はじめに

情報過多にある現状を改善するために多くの情報収集支援技術が開発されている。その手法として、情報そのものをテキスト解析等の技術で機械処理し、整理することで支援するものの他に、情報に関する人の知識を利用することで支援するものがある。後者は機械処理では困難な情報の価値的な部分まで扱える利点があるが、人の協力が必要であり、システムが利用者へ多くの負荷をかけるなどの理由で協力が得られなかった場合には、うまく機能しなくなるという問題がある。そこで本研究ではより利用者が協力しやすい情報共有の形態と考えられる、ネットワーク型の情報共有システムを提案する。

2 情報共有の負荷

人の知識を用いた情報収集支援技術には、参加者の協力が欠かせない。例えば、協調フィルタリングではある程度の量の情報が集まらなければそもそも機能しない。また、メーリングリスト等のオンラインコミュニティにおいても、参加者が活発に議論する姿勢を維

持し続けなければ情報共有の場として機能しない。

では、参加者の協力が得られない状況はどのようにして起こるのだろうか。それは情報共有に対する負荷が、得られるメリットを上回ったときと考えられる。では、情報共有における負荷とはどのようなものがあるのか。今回、以下のような二つの負荷を考えてみた。

- **情報作成の負荷** 情報を提供するだけで、その情報が直接には自分にとって価値の無い物である場合、その情報を作ることに負荷を感じる。
- **情報提供の負荷** 苦勞して自分で作った情報を、見知らぬ他人にまで公開することへの抵抗感が負荷と感じる。

多くの協調フィルタリングシステムでは、情報に対する 5 段階評価といった作成容易なものを共有情報として用いることで負荷軽減を行っている。だが、それは情報作成そのものの価値を減らすことになる。これは、すでに十分に情報が蓄積され、共有によるメリットを情報提供者が得られるような状態ならば問題ないが、それに至らない場合、例えば蓄積された情報量が不十分な初期段階などにおいては、大きな問題となる。

オンラインコミュニティでは、情報作成の負荷は大きくとも、参加意識によって情報提供の心理的負荷を軽減することで、全体として負荷を小さくしている。だが、コミュニティが肥大化して参加意識が希薄になってきた場合、この構図は崩れてしまう。

Proposal of The Collavolative Filltering Networks using Shared Topic Networks between Persons.

Masahiro HAMASAKI, 奈良先端科学技術大学院大学, Nara Institute of Science and Technology (NAIST).

Hideaki TAKEDA, 国立情報学研究所, National Institute of Informatics.

Yasuyuki KONO, 奈良先端科学技術大学院大学, Nara Institute of Science and Technology (NAIST).

Masatsugu KIDODE, 奈良先端科学技術大学院大学, Nara Institute of Science and Technology (NAIST).

3 協調的情報推薦ネットワーク

3.1 人のネットワーク

以上をふまえ、本研究では、共有を前提としたものではなく自分にとって有用な情報を、友人や知人といった人と人の関係で作られる範囲で共有する、強調的情報推薦ネットワークを提案する。共有情報が自分にとって有用な情報ならば作成の負荷は少なく、また、共有範囲が友人や知人などの見える範囲ならば提供の負荷も少ないと考えられる。だが同時に、共有を前提としてない情報をどのような形で共有し、友人関係から作られるネットワークをどのように広げていくか、という問題がある。

そこで、我々は個人の情報を連結する方法として共通話題ネットワークを、ネットワークを広げる方法として仲介者モデルを提案する。前者は話題を用いることで共有時の知識のギャップを埋め、後者は仲介者を用いることで心理的な負荷を感じさせずに共有範囲を拡充する手法である。

3.2 共通話題ネットワーク

共通話題ネットワークとは、人がそれぞれ持つ話題を、共通性を元につなげたネットワークである。利用者は共通性を持った話題（共通話題）を介して他の人と情報共有を行うことが出来る。今回は共有情報として Web ブックマークを用い、そして Web ブックマークを用いた共通話題ネットワーク作成には *kMedia* [Takeda00] を用いる。

Web ブックマークのフォルダは Web ブラウザのブックマークに登録する Web ページを階層構造で整理するためのものである。ユーザは任意にフォルダを作り、その中に記録しておきたい Web ページを登録できる。*kMedia* では、その Web ページが登録されたフォルダを一つの話題とみなし、フォルダに登録している Web ページはそのフォルダ（話題）を示す内容であるとする。

ユーザ同士の話題の共通性は、その話題に含まれる情報がどれだけ類似しているかによって決める。*kMedia* ではブックマークフォルダに含まれる情報、すなわちフォルダ内に登録された Web ページの類似性

から話題の共通性を求める。我々は、この手法が人と人をつなぐ技術として利用可能かどうかを被験者を用いて検証した [Hamasaki01]。

3.3 仲介者モデル

仲介者モデルとは、互いに見知らぬユーザ A と B の共通の知人 C が仲介者となり、お互いを紹介する仕組みである。だが、仲介者によって新しい人を紹介してもらおうとしても、紹介される両者はまだ未知の関係であり、また、新たに関係を結ぶかどうか未定である。そのため、両者には関係を結ぶかどうか判断するのに十分な情報が与えられる必要があると同時に、まだ見知らぬ相手に対して自分を特定できるような情報は与えてはならないという制約がある。

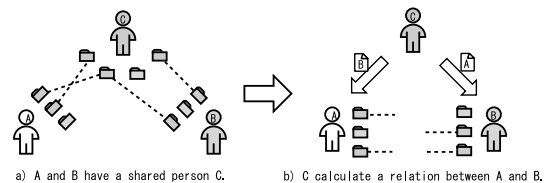


図 1 仲介者モデル

そこで、分類の類似度と、発見された共通話題のみを紹介情報として伝える。この際、相手が持つ話題の名前（＝フォルダ名）は伝えられず、自分の持つ話題で相手と共通性のあるものが提示される。つまり友人を介して知り得た相手からは、関係を結ばない限りは、全く相手を特定できない情報しか得られない。

4 今後の課題

本論文では、ネットワーク型の情報共有モデルを提案した。その有効性を検証するためにも、今後は提案に基づく実装を作成し、被験者を用いて評価実験を行いたい。

参考文献

- [Takeda00] H. Takeda, T. Matsuzuka, Y. Taniguchi: Discovery of shared topics networks among people – a simple approach to find community knowledge from www bookmarks, *Proceedings of the PRICAI00*, 2000.
- [Hamasaki01] M. Hamasaki, H. Takeda: Experimental Results for a Method to Discover Human Relationship Based on WWW Bookmarks, *Proceedings of the KES2001*, 2001.

RoboCupRescueにおける 最少通話による協調の実現

森本 武資 河野 健二 竹内 郁雄

1 はじめに

マルチエージェントシステムの研究では、対象が非常に限定されたトイ的な問題をテストベッドとする場合が多い。トイ的な問題における研究は、問題の本質をとらえやすいが、成果を実際的な問題に適用していくという欠点を併せ持つ。マルチエージェントシステムに関する実際的な問題を研究するためには、様々な事象を対象とするオープンな問題を扱うことが重要である。筆者らは、大規模災害シミュレーションを扱う RoboCupRescue シミュレーション・プロジェクトに参加し、多種類のエージェントが多数活動するマルチエージェントシステムの開発に取り組んでいる。現在、シミュレーションシステムのプロトタイプが開発され、この上で救助活動を行うエージェントの研究が可能になった。しかし、同エージェントの研究が進むにつれ、一見複雑な同システムが実はトイ的な問題に帰着できることが分かり、オープンな問題が本質的に抱える、問題設定と実装の難しさを改めて示す結果となった。本稿では、同システムの概要を示し、筆者らのエージェントの協調を述べ、筆者らのこれまでの成果を紹介する。

2 RoboCupRescue プロジェクト

RoboCupRescue プロジェクトは、大規模災害に対してロバストな社会を創ることを最終目標とするプ

ロジェクトである [1]。現在、シミュレータ上の仮想的な災害空間で活動するエージェントを研究するシミュレーション・プロジェクトと、災害現場で実際に活動するロボットなどを研究するロボティクス & インフラストラクチャ・プロジェクトが進められている。

シミュレーション・プロジェクトが開発している大規模災害シミュレーションシステムは、マルチエージェントシステムのテストベッドとして利用可能な段階に入った。2001 年 4 月に福岡で開催された RoboCupRescue JapanOpen を皮切りに、2001 年 7 月の RoboFesta 関西、8 月の RoboCupRescue2001 in USA において、同システム上で活動するエージェントの救助能力を評価するための競技会が開催された。シミュレーション・プロジェクトは、同システムを発展させることにより、防災情報インフラと結合させ、実際の災害情報とシミュレーションを融合したシステムにし、より有効な意思決定支援に役立てることを目標としている。

3 エージェントの活動目的と災害空間

大規模災害シミュレーションシステムで活動するエージェントには、市民、救急隊、消防隊、道路の寸断を修復する道路啓開隊、通信を行う救急、消防、啓開各隊の司令所の 7 種類がある。エージェントの活動目的は死者数を最小にすることであり、同異種のエージェントが協調することで負傷者を救助する。

同システムは災害発生直後の市街地をシミュレートする。この市街地を災害空間と呼ぶ。死傷の主な原因には埋没と火災が挙げられる。震災直後には家屋

Takeshi Morimoto, 電気通信大学 情報工学科

Kenji Kono, 電気通信大学 情報工学科

Ikuro Takeuchi, 電気通信大学 情報工学科

が倒壊し多数の市民が閉じ込められ負傷する。更に、家屋から逃げられず火に巻き込まれれば、致命的なダメージを被る。救急隊は一刻も早く負傷者を見つけ出し救助しなければならないが、道路は瓦礫などで寸断されており、負傷者の所まで移動できない。負傷者の救出には、まず道路啓開隊が啓開活動により救急隊と消防隊の移動経路を確保し、消防隊が消火活動により負傷者が火に巻き込まれるのを防ぐ必要がある。

エージェントは同異種間で協調する必要があるが、個々のエージェントの視覚能力と通信能力は非常に制限されており、エージェントは限られた情報だけで協調を実現しなければならない。加えて、災害空間はダイナミックに変化する。負傷者は徐々に消耗し、火災は急激に広がり、他のエージェントは動きまわる。つまり、行動の対象自体が時々刻々と変化する。エージェントは、行動対象の変化に応じて、活動と協調のプランを柔軟に変更できなければならない。

4 大規模災害シミュレーションシステム

4.1 構成と処理

シミュレーション・プロジェクトが開発している大規模災害シミュレーションシステムは、複数のモジュールから構成される(図 1)。モジュールには、個々の災害をシミュレートする Sub-simulator、災害空間の地理情報を与える GIS (Geographical Information System)、災害空間の状況を視覚化する Viewer、これらを統合する Kernel がある。個々のエージェントも、システムを構成するモジュールの一つとして位置づけられる。

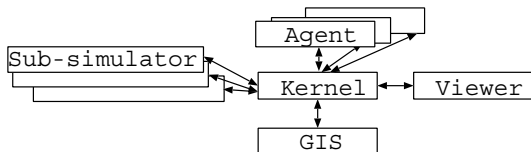


図 1 シミュレーションシステムの構成

災害空間のシミュレーションは 1 分刻みで進行する。災害空間上の 1 分を 1 ターンと呼ぶ。エージェントは 1 ターン毎に、周囲の状況を知覚し、行動を決

定し、それをシステムに伝える。行動は次のターンに反映される。具体的には、以下の繰り返しによって、シミュレーションが進行する。

1. Kernel が個々のエージェントに視覚情報を送る
2. エージェントが Kernel に行動内容を伝える
3. Kernel が行動内容を Sub-simulator に送る
4. Sub-simulator が更新結果を Kernel に伝える
5. Kernel が統合結果を Sub-simulator に送る
6. 災害空間の時間を 1 分進める

エージェントは視覚情報を受けてから 1 ～ 2 秒以内に行動を決定しなければならない。

4.2 災害空間のモデルと問題点

災害空間は建物や道路、人間等のオブジェクトの集合として表現され、個々のオブジェクトは複数のプロパティを持つ。災害空間は 500 m 四方の市街地であり、約 2500 個のオブジェクトによって構成される。

オブジェクトには、一般家屋、避難所、各隊の司令部といった建物オブジェクト、道路とその端点を表すオブジェクト、川とその端点を表すオブジェクト、市民、救急隊、消防隊、道路啓開隊、車といった移動物体オブジェクト、災害空間の場所や風向き等を表す世界オブジェクトがある。しかし、全 15 種類のうち、世界、川、川の端点、車の 4 種類は実質的には使用されない。災害空間上に存在するのは、建物と道路、人間だけである。

建物には 13 個のプロパティがあり、道路には 15 個、人間には 8 個あるが、システムにより実装され、かつエージェントの問題解決に必要なプロパティはわずかである(表 1)。

表 1 問題解決に必要なプロパティ

オブジェクト	プロパティ
建物	位置、燃焼度、面積
道路	位置、幅、閉塞幅、車線数
人間	現在地、生命力、負傷度、埋没度

災害空間の仕様は複雑だが、その複雑さには問題設定を豊かにするという点において有意義とはいえない部分がある。加えて、複雑な仕様に開発が追いつ

表 2 エージェントの能力

種類	能力
市民	Sense, Hear, Move(Route), Say(String)
救急隊	Sense, Hear, Move(Route), Say(String), Tell(String), Rescue(Human), Load(Human), Unload
消防隊	Sense, Hear, Move(Route), Say(String), Tell(String), Extinguish(Building)
道路啓開隊	Sense, Hear, Move(Route), Say(String), Tell(String), Clear(Road)
救急隊司令所	Hear, Say(String), Tell(String)
消防隊司令所	Hear, Say(String), Tell(String)
道路啓開隊司令所	Hear, Say(String), Tell(String)

かず、現在開発中のプロトタイプ・シミュレーションシステムの実装は不安定である。不安定な実装の上では、協調以前に、単体で活動できるエージェントの実現すら難しく、実際、筆者らがエージェント開発に要した労力の大半は、単体行動の実現のために費された。

このような問題設定の難しさや、それに付随するシステムの安定性の問題は、オープンな事象を扱うマルチエージェントシステムが本質的に抱える問題である。RoboCup サッカー [2] でも、我々はこれと同じ問題に直面したが、その際もやはり、協調の前に、まず単体の行動能力の向上が必要であるという認識に立ち、世界トップクラスのサッカー・エージェントを開発した [3]。複雑な問題設定の中では、マルチエージェントの協調について、トイ的な問題をテストベッドとするのとはまったく別の苦労がある。

4.3 エージェントのモデル

個々のエージェントは、数人の人間の集まりを表している。市民は家族を、救急、消防、道路啓開隊は小隊を、司令所はそこで働く人達の集まりである。災害空間上では、72 家族の市民と、5 隊の救急隊、10 隊の消防隊、10 隊の道路啓開隊、そして司令所が各 1 棟活動する。

エージェントは、行為とその対象をシステムに伝えることにより行動を起こす。ある経路 *route* に沿って移動するには、*Move(route)* をシステムに伝える。経路は道路や建物のリストで表現する。同様に、人間 *human* の救助は *Rescue(human)*、救急車への乗せ降ろしは *Load(human)*、*Unload*、火災 *building* への放水は *Extinguish(building)*、道路 *road* の啓開は *Clear(road)* をシステムに伝える (表 2)。

エージェントが情報を獲得する手段には、視覚と通信がある。個々のエージェントは、火災と半径 10m 以内の建物や道路、他のエージェントなどを視覚でき、肉声と電気通信による通信ができる。肉声は話者から半径 30m 以内にいるエージェントに聞こえ、小隊による電気通信は同種の小隊と司令所に伝わる。司令所による電気通信では、他種の司令所にもメッセージが伝達される (図 2)。個々のエージェントの発話

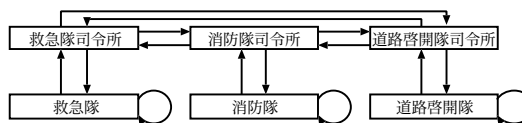


図 2 電気通信の流れ

と聞き取りはそれぞれ 1 ターンに 4 メッセージに限られている。聞き手は、届いたメッセージの中から 4 つを話者によって選別し、内容を聞き取ることができる。1 つのメッセージの情報量は、1 センテンス程度に制限されている。小隊と司令所の通信回数の制限が同じであるのは一見すると不自然だが、実際に東京消防庁に置かれている通報用の電話の台数が 10~20 台程度であることを考えれば、妥当な問題設定である。

しかしながら、エージェントの通信能力の制約は非常に厳しい。メッセージの伝達には 1 ターンの遅延がある上に、肉声は伝えられる対象が限られており、電気通信はメッセージの到達率が低いからである。

遅延は同種の小隊間ならば許容範囲だが、異種小隊間の電気通信の伝達経路は、話者 → 同種司令所 → 異種司令所 → 異種小隊と長く、3 ターンもの遅延が生じる。

肉声ならば、異種小隊間でも遅延を 1 ターンに抑

えられるが、半径 10m の視野では他のエージェントの位置の把握が難しい上に、30m の肉声伝達範囲では声が届く前に移動してしまっている場合が大半である。肉声による通信は、埋没した市民、や司令所のような動かない聞き手に対してのみ有効であるといえる。

電気通信には、話すメッセージが多い程、聞き手がメッセージを聞く確率が低くなるという、通信量と到達率のジレンマがある。例えば、消防隊 10 隊と消防隊司令所 1 棟がそれぞれ 4 回電気通信した場合、次のターンには、各消防隊に 40 ものメッセージが届く。消防隊が内容を聞き取れるのは 4 メッセージだけであるから、届いたメッセージの 10% しか聞き取れないことになる。したがって、重要なメッセージを聞き逃さないためには、各エージェントが、最小限しか話さないようにする必要がある。

5 実現したエージェント

5.1 単体の行動能力の向上

複数のエージェントが協調的活動を行うためには、個々のエージェントが基礎的な行動を確実に遂行できる能力を備える必要がある。RoboCupRescue における基礎能力とは、迅速な移動能力である。筆者らは、通過可能性に基づいた経路探索により到達確率の高い経路を構成することで、迅速な移動を実現した。これには災害空間の以下の性質を利用した。

- 一度通過した経路は渋滞を除き確実に通過できる
- 閉塞は道路毎に判定されるため、未確認道路数が少い程、閉塞遭遇率が低くなる
- 車両は高速に移動できるため、移動距離を考慮しなくてよい

経路構成の方針は、まず通過可能を確認した道路のみで構成を試み、次に未確認道路数最少の経路の構成を目指す。それでも目的地までの経路が見つからなければ、閉塞も含む経路を構成する。このアルゴリズムは、最良優先探索に、以下のコスト関数 (式 1) を与えることで実現できる。ここで、 M は最大経路長より大きな値とする。

$$costFunc(Road) = \begin{cases} 1 & \text{通過可能} \\ M & \text{未確認} \\ M^2 & \text{閉塞} \end{cases} \quad (1)$$

こうして得られた経路上の移動を繰り返すことで、エージェントは閉塞状況をリアルタイムに学習しながら、その時点で知り得る最良の経路を移動する。

5.2 救急隊

死傷者に直接関わる救急隊の活動は非常に重要である。多数の市民が倒壊家屋に埋もれているのに対し、救急隊は 5 隊しかいない。救急隊は、救助活動の効果を最大限に引き出すために、負傷の程度によって負傷者を選別し、生命の危機に瀕している負傷者を最優先に救助する“トリアージ”[4]に基づいて活動する。具体的には、負傷者を以下の 4 段階に分けて救助する。

1. 即救助 … 救助しないと助からない
2. 救助 … 自力で避難はできないが当面生き残れる
3. 救助不要 … 自力で避難できる
4. 救助断念 … 救助しても助からない

救急隊は「即救助」「救助」の負傷者を発見した時のみ救助を始め、「救助不要」「救助断念」の負傷者を発見しても搜索を続ける。同じ場所に複数の負傷者がいる場合には、まず「即救助」の救助を優先する。優先度が同じ場合には、確実のため、埋没度の少ない負傷者の救助を優先する。

負傷者は時々刻々と消耗し傷も深くなるため、迅速な救出活動が必要である。救出にかかる時間を短縮するためには、救助能力を集中させなければならない。救急隊は、「即救助」「救助」の優先度をもつ負傷者を発見した時に、電気通信によって他の救急隊に報告し、報告を受けた救急隊はその場所に駆けつける。複数の報告が同時に届いた場合には、迅速に到達できる場所にいる負傷者の方に駆けつけ、別の負傷者の救助中に報告を受けた場合は、移動せず救助を続ける。この協調により、負傷者を救出できる確率が向上する。

5.3 消防隊

火災は状態が急激に変化するため、特に素早い対応が必要な災害である。ごく初期の火災は 1 隊の消防

個々の消防隊は、火災の状態、火災周囲の状況、火災と自分との距離から、火災の放水優先度を見積り、優先度が高い火災に放水/移動する。図3の数字は優先度を表す。1が最優先であり、これは放水範囲内にあって、消火価値が高く、ごく初期段階の火災を表す。もしこの条件を満たす火災があれば、消防隊はそれに放水する。



```

消火価値 (Fire) = 未着火数 - 火災数 × 減少率
if 未着火数 > 0
    then 未着火数 += 鎮火数

```

$$costFunc(Road) = \begin{cases} 1 & \text{通過可能} \\ M & \text{未確認 or 閉塞} \end{cases} \quad (3)$$

6 実験

上述の 3 つの協調行動，救助能力の集中，閉塞と啓開の報告，放水能力の分散について，その有効性を調べるために実験を行った．実験を行った災害空間では，震災直後に 4 棟が出火し 33%の道路が閉塞する．

表 3 協調の有無と活動結果

集中救助	閉塞報告	放水分散	死者数	火災数	鎮火時間	救出人数
しない	しない	しない	25	141	203	32
する	する	しない	25	53	167	34
する	しない	する	26	32	178	32
しない	する	する	24	21	81	29
する	する	する	23	10	41	37

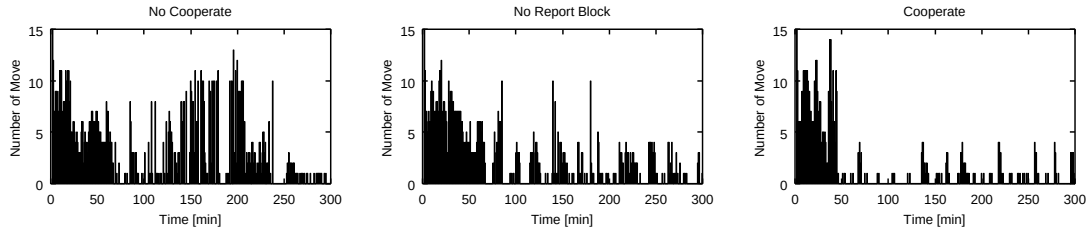


図 4 協調の有無と移動回数

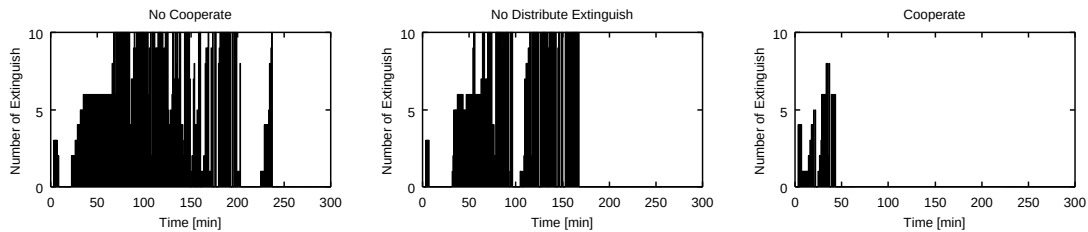


図 5 協調の有無と放水回数

実験結果を図 3 に示す。

救助能力を集中した方が、しない場合よりも多くの負傷者を救出できている。閉塞と啓開を報告した場合と、放水能力を分散した場合も、しない場合より速く鎮火が完了している。また、3つの協調行動を全て行うことで、効果的な救助活動、消火活動を実現している。

閉塞報告と放水分散の働きを詳しく見るために、協調の有無による時間毎の移動回数（図 4）と放水回数（図 5）を示す。

図 4 は、左から順に、協調しない場合、啓開報告をしない（他は行う）場合、3つの協調を行う場合の、救急隊と消防隊の、時間毎の移動回数を示している。協調しない場合は鎮火が完了するまで、多数のエージェントがほぼ絶え間なく移動し続けている。移動の多さは、目的地に到達するまでに時間がかかっている

ことを示している。啓開報告をしない場合（中央の図）は、放水分散によって無駄な移動を抑えているため、移動するエージェントが少ない。全ての協調を行う場合、消防隊は初期段階で鎮火を完了する。それ以後の移動は、救急隊が次の負傷者の場所に向かうための周期的なものがほとんどである。

図 5 は、協調しない場合、放水分散しない場合、全ての協調を行う場合の時間毎の放水回数を示している。協調しない場合は、初期段階に放水できず、火災が燃え広がった頃になってようやく放水できるようになっている。放水分散しない場合も初期消火に失敗しているが、閉塞報告によって火災に近けるため 100 ターンあたりから全ての消防隊が放水できている。全ての協調を行う場合は、消防隊は初期消火に成功し、わずかな放水で鎮火を完了している。これは放水分散によって無駄な移動を抑えられているためである。

火災に移動する際も、他に移動するエージェントが少いため、啓開隊によるサポートを受けやすくなる。

以上により、実装した協調の有効性が確かめられた。

7 競技会での評価

RoboCupRescue プロジェクトは、大規模災害シミュレーションシステム上で活動するエージェントの救助能力を評価するために、競技会を開催している。筆者らが開発したエージェント YabAI は、これまでに開催された全ての大会、RoboCup JapanOpen (2001 年 4 月), RoboFesta (2001 年 7 月), RoboCup2001 in USA (2001 年 8 月) で優勝した。米国で開かれた第一回世界大会 (4ヶ国 7 チームが参加) ではイランのチーム (Arian) と接戦を繰り広げたが、それを除けば他を圧倒していたといえる。

YabAI の最大の勝因は、個々のエージェントの高い基礎行動能力にあり、本題となるべき個々のエージェントの自律や複数エージェント間の協調というテーマにあまり触れていない。この結果は、現在の大規模災害シミュレーションシステムが、エージェントの分散協調の研究以前の段階にあることを示している。

8 まとめ

様々な事象を対象とするオープンな問題を扱うために、RoboCupRescue シミュレーションプロジェクトが開発中の大規模災害シミュレーションシステム上で活動する防災救助エージェントについて研究を行った。しかし、一見複雑な同システムが実はトイ的な問題に帰着できることが分かり、オープンな問題が本質

的に抱える、問題設定と実装の難しさを改めて示す結果となった。協調について若干の知見が得られたが、今後は、問題自体、Sub-simulator 群のさらなる改善が必要である。しかしその分、問題自身がオープンであるといえる。このような問題はマルチエージェントの協調について、トイ的な問題をテストベッドとするのとはまったく別の苦労があることを報告した。RoboCup サッカーでもこのようなオープン性はあったが RoboCupRescue ではオープン性が桁違いに大きい。しかし、より現実に即した問題であり、このようなある種、黎明期の混乱の中、このような複雑な (怪奇と付け加えてもよいくらいの) 問題設定の中でも、エージェントの協調について、我々の RoboCup サッカーの研究と同様、問題が複雑であれば協調の前にまず個々のエージェントのスキルの向上が必要であるという一般的な知見が得られたということは結論できるであろう。

参考文献

- [1] 田所, 北野編: ロボカップレスキュー: 大規模災害救助への挑戦, 共立出版 (2000)
- [2] P. Stone, T. Balch, G. Kraetzschmar (Eds.): RoboCup 2000: Robot Soccer World Cup IV, lecture notes in Computer Science. Springer Verlag, Berlin, 2000.
- [3] 鈴木隆志, 栗田英明, 竹内郁雄: RoboCup シミュレーションエージェントの能力とワールドモデル, 情報処理学会研究報告, Vol. 2000 No. 27(GI-2) pp. 25-32, (2000)
- [4] 高橋友一他: 大規模災害におけるマルチエージェントシステム, MACC2000 (2000)
- [5] Peter Stone and Manuela Veloso: Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork, Artificial Intelligence (AIJ), 1999.

異種端末間の協調による高齢者・障害者の移動支援, Robotic Communication Terminals

矢入(江口) 郁子 猪木 誠二

1 はじめに

人間にとって「移動」とは、自立のかつ快適に生活するための手段として極めて基本的、かつ必要不可欠な行動である。しかし現在の社会環境下では、視覚・聴覚・下肢駆動機能に障害を持つ高齢者・障害者の場合、移動に不可欠な認知・駆動・情報入手の3つの要素行動に問題が生じるため、自立的な移動が困難となっている。そこで筆者らは、幅広い高齢者・障害者を対象に、障害の種類、レベル、障害歴の違い、重複障害などの多様性を考慮に入れ、認知・駆動・情報入手の3つの要素行動を補助する移動支援システム, Robotic Communication Terminals(以後, RCTと略記)の実現を目指し、研究を行っている[矢入 01]。これまで高齢者・障害者の移動支援の研究では、携帯電話や自律走行ロボットを使った視覚障害者のナビゲーションシステム[東海 99][Kemmerling 98][小谷 97]や、肢体障害者のための知能化車椅子[足達 99]など、高齢者・障害者のうちの一部のユーザを対象に、移動の問題の一部の解決を試みる研究は数多くなされてきた。一方、健常者を主対象にした研究では、認知・情報の入手の補助を行う歩行者用 ITS の実現が国の政策として位置付けられ[国土交通省 01]、産業界では携帯電話や環境に埋め込むビーコンを用いた早期実

現可能なシステムの開発が活発化している。RCTではこれらのシステムの先を見据え、環境やユーザのモニタリング、異種端末間の協調などの技術を用いた、より広い観点での認知・駆動・情報の入手の補助方法を提案し、歩行者用 ITS の高度化に貢献することを目指す。

本稿では、RCTの概要、研究目標を示し、マルチエージェント研究の観点からみたRCTの位置付けを述べる。そして最後に、筆者らのこれまでの成果および今後の展開を紹介する。

2 Robotic Communication Terminals

2.1 RCT による移動支援

RCTは、以下の3つのタイプの端末からなる。

<環境端末> アンテナのような役割を持ち、道路や駅などに設置される端末である。それぞれの環境端末は設置された場所の周辺を監視し、障害物や人・車の有無などの環境に関する情報を検出する。環境端末が検出した情報は、周辺を移動中のユーザに伝えられるだけでなく、遠隔地のユーザにも移動経路を決めるための材料として提供される。

<ユーザ携帯型移動端末> 携帯電話やPDAを進化させた端末である。この端末を持ち歩くユーザはインターネットを通じて目的地への経路案内や交通機関の事故などの情報を入手する。そしてさらに環境端末との通信によって、周辺や遠隔地の障害物や人・車の有無などの情報を入手することも可能である。これらの情報は、ユーザの身体の状態に合わせて加工され、ユーザの身体の状態に合わせて設計された画像・

音声・触覚を利用したインタフェースを介して提示される。

<ユーザ搭乗型移動端末> 高齢者・障害者のための乗り物を進化させた端末である。ユーザ携帯型移動端末の機能に加え、ユーザの操縦通りに運転できる通常の乗り物としての機能、さらにセンサー及び環境端末からの情報を利用した障害物の自動回避などの運転支援機能も装備される。また、この端末にはユーザの身体の状態に合わせて設計されたハンドルやジョイスティックなどの操縦系、電動のスクータや車椅子のボディが用いられる。

これらの役割の異なった端末同士が通信し互いに協力しあうことで、認知・駆動・補助の 3 つの要素行動を補助し、ユーザの市街地での移動を支援する。図 1 に RCT による移動支援のイメージを示す。例えば移動端末は、移動中のユーザから離れた場所をモニタする環境端末の「こっちからは何も来ないよ」というメッセージを近傍の環境端末を通して入手し、ユーザに「前からは何も来ません」という情報を伝えることができる。また、自転車を検出した環境端末は、移動方向や位置から危険と判断した近傍の移動端末に、「自転車きたよ、ぶつかる可能性があるよ。」というメッセージを送る。それを受けた移動端末は、例えばユーザ搭乗型であれば自動停止してやり過ごすなどの補助を行ったり、ユーザに危険情報を伝えることができる。

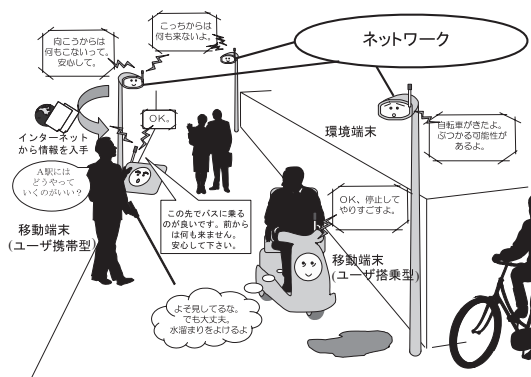


図 1 RCT による移動支援のイメージ

2.2 RCT の目標

RCT では、5 年以内に以下の目標 1~3 の要素技術を実現し、システムの実用化への基礎を築くことを目指している。

<目標 1> ユーザの多様性の適切な分類方法と、その分類に基づいた支援タスクおよび移動端末のハードウェアデザイン、移動端末のソフトウェアのユーザへの適応方法

RCT では高齢者・障害者の多様性を考慮した支援タスクを、個々のユーザに適した方法で実行する移動支援を目指す。具体的には (1) 移動に不可欠な認知・駆動・情報入手の 3 つの要素行動と身体の状態との関係に注目し、ユーザを 20~30 個程度のグループに分類、(2) そのグループ毎に支援タスク、移動端末のハードウェアのデザインの指針を設定、(3) 移動端末のソフトウェアが、入力された身体の状態や嗜好などに合わせて動作したり、使い込むうちにユーザの癖や習慣に合わせて適応する仕組みを実現する。

<目標 2> 環境端末、移動端末およびユーザ間の協調によって、移動支援タスクを場面、状況、ユーザの状態に合わせて実行するためのシステムアーキテクチャ

環境端末、ユーザ搭乗型・ユーザ携帯型移動端末の 3 つのタイプに分類される多様な端末からなる分散協調システムとして全体システムを構成する。また、RCT のサブシステムとしての環境端末および移動端末を、環境やユーザとインタラクションし適応するロボット的なシステム、すなわちロボティックシステムとして構成する。そして、移動端末とユーザとの密なインタラクションによって、RCT の全体システムとユーザ間の協調を実現する。これらのシステムアーキテクチャをプロトタイプシステムの製作を通して示す。

<目標 3> 実世界、ネットワーク世界からの情報の取り込み・加工・蓄積、およびその情報の伝達方式

RCT では、認知・駆動・情報の入手の移動の要素行動の補助のうち、情報の入手の補助に最も注目している。「実世界」、「インターネット」の 2 つをユーザに提供する情報のリソースとし、「専用の情報提供用サーバの構築と利用」、「既存のインターネット上の多

数のサーバ上の情報の有効利用」の 2 つの方法による情報の入手の補助の具体的なアプリケーションを提案し、プロトタイプを作成する。そして、2 つの情報リソースからユーザに情報を提供するまでの、取り込み・加工・蓄積・伝達の方式を示す。

2.3 マルチエージェント研究としての RCT

RCT は以下の 3 つの種類のエージェントから構成されるマルチエージェントシステムとして捉えることができる。

- (1) **端末内部エージェント**: 異なる機能を持ち、群として環境端末、移動端末の単体端末の内部を構成する。
- (2) **端末間通信エージェント**: 個々の環境端末、移動端末を代表し、端末間の通信を担い、協調動作を実現する。
- (3) **外部通信エージェント**: インターネットなどのネットワーク上に構築された他のシステムとの通信を担い、協調動作を実現する。

認知・駆動・情報の入手の補助タスクがどの種類のエージェントによって実行されるかを表 1 に示す。

表 1 RCT の 3 つの種類のエージェントと補助タスクの
実行 (※はユーザ搭乗型移動端末のみ)

補助タスク	(1) 端末内部 エージェント	(2) 端末間通信 エージェント	(3) 外部通信 エージェント
認知	○	○	—
駆動	○※	—	—
情報の入手	○	○	○

認知の補助タスクに関しては、端末内部エージェントが端末の周囲の環境の認識を行い、端末間通信エージェントが認識結果を近傍の端末に送り届けるという役割分担がなされる。また情報の入手の補助タスクに関しては、端末内部エージェントがユーザの要求の解釈および情報の加工・提供を行い、端末間エージェントが個々の端末に蓄積された実世界データを探索・獲得し、外部通信エージェントがインターネットなどのネットワーク上に構築された他のシステムに蓄積されたデータを探索・獲得するという役割分担がなされる。

端末内部エージェントのように、機能の異なったエージェント群で一つの知的システムを構築するアプローチは、古くから AI およびロボットの分野で研究がなされてきた。また端末間通信エージェントのように、複数の端末間の協調によってタスクを実行するアプローチは、群ロボットによるタスク実行の研究や[Balch 94]、端末間を移動するエージェントによる通信の研究[佐藤 00] などがなされている。そして外部通信エージェントのように、インターネット上のサーバのデータから有効な情報を探索・獲得し提供するアプローチは、サーチアルゴリズムの研究や[Levy 96]、データマイニングの研究[Cooley 97] などの多数の研究がなされてきた。しかし、これらのアプローチの全てを含む統合的なマルチエージェントシステム研究は例を見ない。RCT は人間の実世界での移動を支援するこれまでにない複雑なマルチエージェントシステムの枠組を示す一つのテストベッドであると言える。

3 筆者らの研究の現状

研究開始から現在までの約 2 年の間の成果と今後の計画を述べ、研究の現状を紹介する。

3.1 ユーザの調査

2.2 の目標 1 のユーザの多様性の分類の課題への取り組みとして、以下の (a)～(c) に示す調査を行ってきた。

(a) 高齢者・障害者の身体の状態と移動の問題との関係の予備調査

身体の状態と移動問題の関係を具体化するため、高齢者および視覚・聴覚・下肢駆動機能に障害をもつ障害者、計 13 名を対象にヒアリングを実施した[Yairi 00]。その結果、視覚障害者は未知の環境・人のいない環境・変化が多い環境での移動に、聴覚障害者は視野外から接近する動物体の認識、アナウンス情報の入手に、そして下肢駆動機能障害者は転倒防止のための歩行面の詳細な状態認識、バリアフリー情報の現場での探索による入手に、それぞれ問題が集中することがわかった。また、障害が重く障害者となった年齢が高いほど、問題の危険度や頻度が増すこと、および使用する補助器具や障害の種類に固有の特徴

的な問題の存在も明らかになった。高齢者の場合は、問題点を絞り込むことは困難であったが、前述した障害者の抱える問題のいずれかに分類されることが多かった。また、加齢により情報の入手を面倒と感じるようになり外出の意欲が低下してきた、という訴えが特徴的であった。

(b) 身体の状態と移動の問題との関係からのユーザ分類の可能性の予備調査

(a) の結果をもとに身体の詳細な状態の質問と、移動の問題を 5 段階の数値で回答する質問からなる調査表を設計し、アンケート調査を実施した。調査表は首都圏の高齢者・障害者団体を通じて配布され、計 698 名の回答を得た。障害の種類、レベル、使用する補助器具、性別、障害の先天性/後天性に関する身体の状態の分類ごとに、移動の問題に関する質問の回答の平均を比較したところ、分類項目と回答の間に強い相関関係の認められる質問や、ある分類項目を境に回答に有意差が認められる質問の存在が明らかになった。このことにより、障害の種類、レベル、使用する補助器具、性別、障害の先天性/後天性をパラメータとして、抱える移動の問題の違いからユーザをグループに分類可能であること、およびパラメータからユーザの抱える移動の問題を推定可能であることが確かめられた。

(c) ユーザ分類のための第 1 次調査

(b) の結果を踏まえ、ユーザ全体を大まかに分類するために全員に同一の調査表を用いてアンケートを実施する第 1 次調査と、さらに詳細なグループにユーザを分類するためにグループ毎に異なった調査表を用いてアンケートを実施する第 2 次調査を計画した。現在は第 1 次調査を実施中である。(b) では、身体障害者の約半数が 65 歳以上であるという現状から、障害者の中年層、若年層の回答数が不十分となり、年齢や障害歴による分類の可能性を確認することはできなかった。そこで今回の調査では中年層、若年層も含めた十分な数の回答を得るため、東京、大阪、京都の 3 大都市の高齢者・障害者団体を通して調査表を配布し、2001 年 10 月現在で約 3500 名の回答を得た。現在は、移動の問題の生じるメカニズムを表現するモデルを決定するための共分散構造分析や、グループに分類するための因子分析などを行っている。ユーザ分類

のためのデータ解析はクラスタリング問題の一種とみなすこともでき、従来手法[Cheeseman 88][Fisher 87]を適用してみる予定である。

3.2 研究インフラの構築

2.2 の目標 2 および 3 に取り組むための研究インフラとして、以下の (a)~(c) に示すプロトタイプシステムを作成した。

(a) 画像による道路監視システム

環境端末の道路監視機能のプロトタイプとして、画像による道路監視システムを製作した。このシステムは、図 2 に示す屋外に設置するカメラ、デュアル CPU の PC、キャプチャボードと画像処理ライブラリから構成される。そして、動物体の形状、位置、移動速度を検出し、歩行者や自転車、自動車などの動物体の種類や、動物体→静止する障害物→動物体という状態変化の判別を行う[吉水 01]。さらに移動端末の位置情報を通信で受け、検出した動物体との位置関係から、動物体を危険物体と判断し、移動端末へ通知することが出来る。道路監視システムの動作の様子を図 3 に示す。下部右側の GUI では監視すべき道路の部分として自動的に切り出された領域の画像が、下部左側の GUI では道路上の動物体として検出された部分の画像が表示されている。そして上部の GUI では検出された動物体が四角形で囲まれ、人、自動車、自転車などの種類の判別結果が文字で示されている。本システムの現在の処理速度は、1 フレーム中の動物体の数が 10 個以下の状態で毎秒 5 フレームである。また性能は、晴天の日中かつ無風状態であれば、移動端末へ



図 2 道路監視システムの屋外設置カメラ 2 基の外観

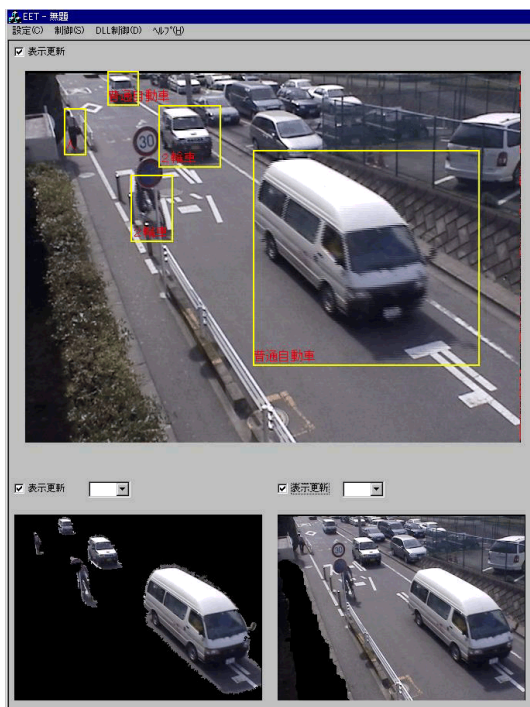


図 3 道路監視システムの動作の様子

の安定した危険物体の通知が可能である[香山 01].

今後は、処理の高速化と、天候の条件の緩和、夜間への対応などの工夫を重ねながら、画像以外のセンサの導入による環境端末のアーキテクチャの提案、実世界情報の蓄積・ユーザへの伝達方式の実現などの課題に取り組む予定である。

(b) 高齢者向けユーザ搭乗型移動端末

移動端末のプロトタイプの一つとして、足腰が少々弱った高齢者向けユーザ搭乗型移動端末、Intelligent City Walker (以後、ICW と表記) のハードウェアを製作した。図 4 に全体像を示す。ICW には、障害物を検出するための超音波・赤外線センサ、障害物回避や操縦補助のための制御装置、他の端末との通信やインターネット接続のための無線 LAN、ユーザと端末との対話のための装置としてのタッチパネルが搭載されている[南山 01]。ICW は他の端末との通信機能、ユーザとの対話機能、ユーザの運転補助機能を持つセミオートノマススクータである。現在は運転補助機能を中心に全機能のインプリメントを行っている。図 5 に目の前に飛び出してきた障害物(段ボー

ル箱) を回避する屋外での実験の様子を示す。搭乗者は回避動作をより効果的にデモするためにハンドルから手を離している。また、見通しの悪い交差点で環境端末から自動車との衝突の可能性の通知を受け、交差点手前で自動停止し、通知された自動車をやりすごす実験も行なった。さらにこれらの実験に加えて、高齢者の試乗による評価実験[田中 01] を行い、山口きらら博などの展示会での一般人の反応とあわせて、ユーザインタフェースの設計に活用している。

今後はセンサやユーザとの対話用装置の改良を重ねながら、運転補助機能の性能の向上、対話インタフェースのデザインの向上を行い、移動端末のシステムアーキテクチャの提案の課題に取り組む予定である。

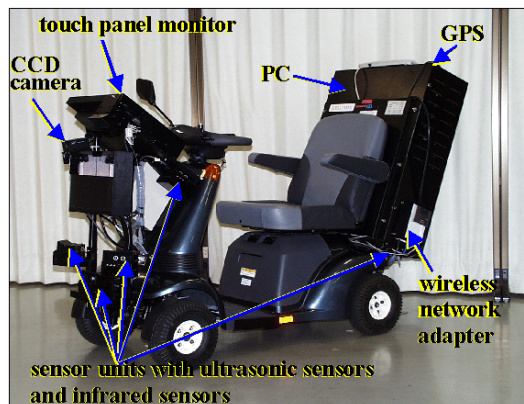


図 4 高齢者向けユーザ搭乗型移動端末 ICW

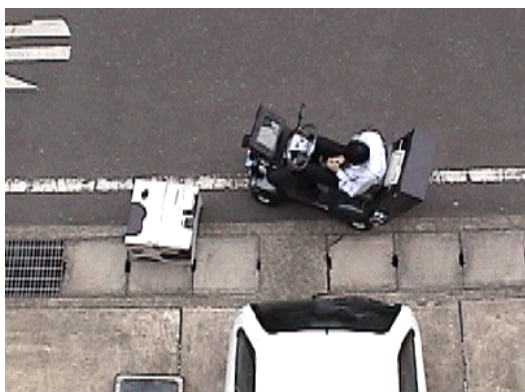


図 5 屋外での障害物回避実験の様子

(c) バリア・バリアフリー情報地図データベース

実世界の情報を専用のサーバを用いてインターネット経由で発信し、ユーザの情報の入手を補助するための具体的アプリケーションとして、歩行者用道路に関するバリア・バリアフリー情報地図データベースのプロトタイプを製作している。このデータベースは、東京都小金井市 (国分寺市の一部を含む) の 500 分の 1 の高精度の地図に、これまでに実現されていなかった歩行者用道路ネットワークを付加し、段差などのバリア情報、盲人用信号などのバリアフリー情報を道路ネットワークにリンクさせて埋め込んだものである[桑原 01][吉岡 01]。図 6 に道路ネットワークの概要を示す。道路ネットワークは歩行者用道路を表すライン、ラインの始点・終点としてのノード、ラインの近傍に存在する停留所などの施設を表しラインにリンクされるポイント、の 3 種類のオブジェクトから構成される。図 7 に埋め込む情報の種類、項目をライン、ノード、ポイントの階層構造から示す。これらの埋め込む情報の種類、項目は、視覚・聴覚・下肢駆動機能に障害をもつ障害者・高齢者、および障害者・高齢者支援団体の指導者の計 14 人に、詳細なヒアリングを実施して決定された。また地図データベースに蓄積した情報は、実際に現地を踏破して収集された。

現在は、2002 年中にインターネット上で一般に公開することを目指し、表示やルート検索用インタフェースを製作中である。今後は身体の状態に適したルート of 検索やナビゲーション文を自動生成するソフトウェアの実現を試みながら、道路監視システムとのドッキングによって、データベースに実世界情報を蓄積する課題にも取り組む予定である。

4 おわりに

本稿では、異種端末の協調によって高齢者・障害者の自立的移動を支援する Robotic Communication Terminals を紹介した。そして、マルチエージェント研究としての位置付け、筆者らの取り組みの現状や今後の展開を述べた。

RCT による移動支援は、健常者を対象とした支援にも容易に適用可能であると期待される。それに関わらず高齢者・障害者の支援に限定して研究対象と

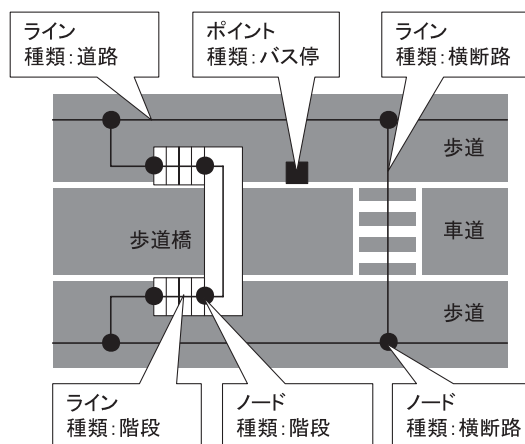


図 6 歩行者用道路ネットワーク概要

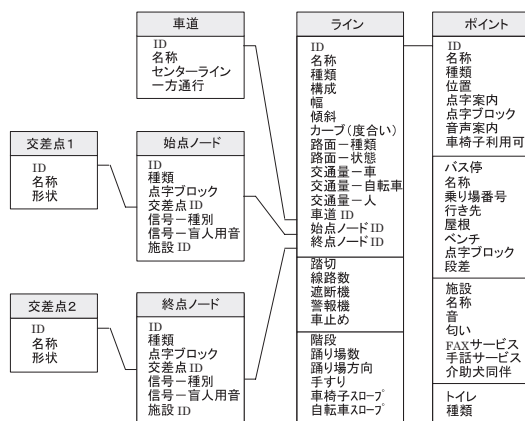


図 7 バリア・バリアフリー情報 DB 概要

する理由には、高齢者・障害者を健常者とは異なるグループとして切捨てて支援システムを実現し、後付けでバリアフリーの実現を目指す従来の手法と異なり、まず高齢者・障害者の立場から網羅的な支援方法を考え、高齢者・障害者の延長線上に健常者が位置すると捉えることこそが、真の意味でのバリアフリーのつながるという筆者らの信条が挙げられる。

また RCT は、このような社会的貢献の面だけでなく、画像・音声認識、プランニング、モバイルエージェント、情報検索、データマイニングなど、マルチエージェント以外にも多数の AI 研究のトピックを含む学術的研究対象であるという一面も持つ。筆者らの取り組みは、これらのトピックの研究を行うための一揃いの研究インフラの構築が完了した段階にあると

言える。今後は、製作中のプロトタイプシステムの完成度を高めながら、ソフトウェアの実装を中心に AI 研究のトピックに深く切り込み、学術的な貢献を目指したい。

参考文献

- [足達 99] 足達, 中西, 他: 顔の視覚情報処理を用いた知的車椅子. 日本ロボット学会誌, vol.17, no.3, pp.113-121 (1999).
- [Balch 94] Balch, T. and Arkin, R. C., :Communication in Reactive Multiagent Robotic Systems. Autonomous Robots, 1, pp.1-25(1994).
- [Cheeseman 88] Cheeseman, P., et al. : Autoclass : A bayesian classification system. In Proceedings of the Fifth International Workshop on Machine Learning, pp. 54-64 (1988).
- [Cooley 97] Cooley, R., et. al. : Web Mining: Information and Pattern Discovery on the World Wide Web, Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (IC-TAI'97), November 1997.
- [Fisher 87] Fisher, D., H. : Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning, 2, pp.139-172(1987).
- [香山 01] 香山, 吉水, 他: 高齢者・障害者の移動を支援するロボティック通信端末のための一般道路監視システム. 第 19 回日本ロボット学会学術講演会, 1M23 (2001).
- [Kemmerling 98] Kemmerling, M., et al. : An Orientation and Information System for Blind People based on RF-Speech-Beacons. TIDE third congress, [http : //www.dinf.org/tide98/tide98.htm](http://www.dinf.org/tide98/tide98.htm), Helsinki, July (1998).
- [国土交通省 01] 国土交通省: ITS [http : //www.its.go.jp/ITS/j-html/index/indexHBook.html](http://www.its.go.jp/ITS/j-html/index/indexHBook.html) (2000).
- [小谷 97] 小谷, 清弘, 森: 視覚障害者のための歩行ガイドロボットの開発. 映像情報メディア学会誌, Vol.51, No.6, pp.878-885 (1997).
- [桑原 01] 桑原, 矢入, 他: 高齢者・障害者向け移動支援 GIS. AI 人工知能学会全国大会, 3A1-05 (2001).
- [Levy 96] Levy, A. Y., et. al. : Querying Heterogeneous Information Sources Using Source Descriptions. Proceedings of the Twenty-second International Conference on Very Large Databases, Bombay, India, pp.251-262(1996).
- [南山 01] 南山, 永合, 他: 高齢者移動支援のためのユーザー搭乗型端末 (IWC: Intelligent City Walker). AI 人工知能学会全国大会, 3A1-04 (2001).
- [佐藤 00] 佐藤, :モバイルエージェント. コンピュータソフトウェア, Vol.17, No.2, pp45-54(2000).
- [田中 01] 田中, 矢入, 他: 高齢者移動支援のためのユーザー搭乗型移動端末 (ICW: Intelligent City Walker) - 運転補助機能実装のためのユーザー評価実験. ヒューマンインタフェースシンポジウム 2001, pp.481-484 (2001).
- [東海 99] 地域振興のための電波利用に関する調査研究会: 視覚障害者を支援する情報通信システムに関する調査研究報告書, 東海電気通信監理局電波監理部企画課, March (1999).
- [Yairi 00] Yairi, I. E. and Igi, S.: A Self-sustained Moving Support System for Aged and Disabled Persons. In Proc. The 6th International Conference on Intelligent Autonomous Systems, pp.692-697 (2000).
- [矢入 01] 矢入, 猪木: 高齢者・障害者の自立的移動を支援する Robotic Communication Terminals(RCT). 人工知能学会誌, vol.16, no.1, pp.139-142 (2001).
- [吉水 01] 吉水, 矢入, 猪木: 歩行者支援のための外乱を考慮した道路監視システム. AI 人工知能学会全国大会, 3A1-02 (2001).
- [吉岡 01] 吉岡, 矢入, 他: バリア・バリアフリー情報を蓄積・利用した移動支援 GIS. ヒューマンインタフェースシンポジウム 2001, pp.479-480 (2001).

未来状態の予測を利用したマルチエージェント強化学習法

長行 康男 柴田 直樹 伊藤 実

1 まえがき

マルチエージェント系における適応行動の実現は、工学及び認知科学の観点から興味深い課題である。その中でも、学習による適応行動の自律的獲得に関する研究が、強化学習 (RL) [1] の発展を契機として近年注目を集めている。

マルチエージェント系において、個々のエージェントが実行する行動の善し悪しは、他エージェントが実行する行動に依存する。この点に注目したマルチエージェント強化学習 (MARL) 法 (マルチエージェント系における強化学習法) として、他エージェントの政策 (行動決定関数) を推定しながら学習を進行する手法が提案されている [2][3][4]。これら 3 つの MARL 法では、推定した他エージェントの政策を基に他エージェントの未来の行動を予測し、その予測行動を考慮に入れながら学習を進行する。これらの MARL 法は、シングルエージェント系を対象として提案された RL 法である Q 学習 [5] を基盤にしている。そして、学習時に他エージェントの政策 (を基に実行される行動) を考慮に入れるため、学習関数である Q 関数を、状態 s と行動 a の関数 $Q(s, a)$ から、状態 s とすべてのエージェントの行動 $a^1, \dots, a^n$ の関数 $Q(s, a^1, \dots, a^n)$ に拡張している。しかしながら、このように Q 関数を拡張した場合、学習関数はすべてのエージェントの行動に依存するため、学習空間サ

イズはエージェント数の増加に対して指数関数的に増加する。強化学習において、学習空間サイズの指数関数的増加は『次元の呪い』と呼ばれ、学習を大幅に遅らせる原因となる。

本稿では、他エージェントの政策を推定しながら学習を進行する MARL 法で、エージェントの行動に依存しない学習関数を利用した新たな MARL 法を提案する。提案する MARL 法は、TD 学習 [6] を基盤にしたもので、学習関数として V 関数 $V(s)$ を用いる。提案する MARL 法では、他エージェントの政策推定に加えて、状態遷移確率関数の推定を行う。そして、推定した状態遷移確率関数と推定した他エージェントの政策を基に未来の状態を予測し、その予測状態における V 関数値を利用しながら強化学習を進行する。

本研究では、マルチエージェント系のモデルとして、先行研究 [2][3][4] と同様、マルコフゲームの枠組みを採用する。

本稿では、提案する MARL 法と拡張 Q 関数 $Q(s, a^1, \dots, a^n)$ を利用した MARL 法 [4] を、マルコフゲームの枠組みでモデル化した 2 体エージェント問題と 3 体エージェント問題に適用する。そして、それらの実験結果を比較することにより、提案手法を評価する。

2 マルコフゲーム

マルコフゲームは、マルチエージェント環境における行動決定問題のモデルで、組 $\langle n, S, A^1, \dots, A^n, T, R^1, \dots, R^n \rangle$ で定義される。ここで、 n は環境内に存在するエージェントの数、 S は環境状態の有限集合、 A^k ($k = 1, \dots, n$) はエージェント k

A multi-agent reinforcement learning method with the estimation of future states.

Yasuo NAGAYUKI, Naoki SHIBATA, Minoru ITO,
奈良先端科学技術大学院大学, Nara Institute of Science
and Technology

の行動の有限集合, T は環境状態の遷移確率関数, R^k はエージェント k の報酬関数である.

各離散時間ステップ $t = 0, 1, 2, \dots$ において, エージェント k ($k = 1, \dots, n$) は, 現在の状態 $s_t \in S$ を観測し, 行動 $a_t^k \in A^k$ を実行する. そして, 状態は $s_{t+1} \in S$ に遷移し, エージェント k は環境から直接報酬 r_{t+1}^k を受け取る. 状態の遷移は状態遷移確率関数 T に従う. この関数は時不変な状態遷移確率

$$T(s'|s, a^1, \dots, a^n) =$$

$$\Pr(s_{t+1} = s' | s_t = s, a_t^1 = a^1, \dots, a_t^n = a^n) \quad (1)$$

の集合で表される. ここで, $\Pr(s'|s, a^1, \dots, a^n)$ は, 状態 s でそれぞれのエージェントが行動 $a^1, \dots, a^n$ を実行したときに状態が s' へ遷移する確率を表す. エージェント k が環境から受け取る直接報酬 r_{t+1}^k も確率的で, その期待値は報酬関数

$$R^k(s, a^1, \dots, a^n) =$$

$$E\{r_{t+1}^k | s_t = s, a_t^1 = a^1, \dots, a_t^n = a^n\} \quad (2)$$

で表される. ここで, $E\{r^k | s, a^1, \dots, a^n\}$ は, 状態 s でそれぞれのエージェントが行動 $a^1, \dots, a^n$ を実行したときにエージェント k が受け取る直接報酬 r^k の期待値を表す.

マルコフゲームにおいて, エージェント k の目的は, 式 (3) で表される関数を最大にするような自分 (エージェント k) の政策 π^k を見つけることである.

$$V^{\pi^k}(s) =$$

$$E\left\{\sum_{n=0}^{\infty} \gamma^n r_{t+n+1}^k | s_t = s, \pi^1, \dots, \pi^n\right\} \quad (3)$$

ここで, 政策 π^k は各状態において各行動を選択する確率への写像 ($\pi^k : S \times A^k \rightarrow [0, 1]$), $\gamma \in [0, 1]$ は割引率と呼ばれるパラメータである. $\pi^1, \dots, \pi^n$ は, それぞれエージェント $1, \dots, n$ の政策を表す. 式 (3) の右辺は, それぞれのエージェントが時刻 t 以降の行動を政策 $\pi^1, \dots, \pi^n$ に従って選択したときに, エージェント k が受け取る割引報酬の和の期待値を表す. V^{π^k} は (エージェント k の) V 関数と呼ばれる. 以下, V^{π^k} を V^k と書く.

3 マルチエージェント強化学習

本稿では, マルコフゲームにおける新たなマルチエージェント強化学習 (MARL) 法を提案する. ここで, 本

研究で取り扱うマルコフゲームは, 不完全情報ゲーム [7] を仮定し, 以下の3つの条件を満たすものとする.

- 両エージェントが同期して行動を実行する.
- 両エージェントがお互いの行動を観測できる.
- エージェント間のコミュニケーションは存在しない.

マルコフゲームにおいて, エージェント k の V 関数はすべてのエージェントの政策に依存している. これは, エージェント k の目的 (V 関数の最大化) が, 他エージェント (エージェント k 以外のエージェント) の政策に依存していることを意味している. この点に注目した MARL 法として, 他エージェントの政策を推定しながら学習を進行する手法がいくつか提案されている [2][3][4]. これら3つの MARL 法では, 推定した他エージェントの政策を基に他エージェントが未来に実行する行動を予測し, その予測行動を考慮に入れながら学習を進行する. これらの MARL 法は, シングルエージェント環境を対象として提案された RL 法である Q 学習 [5] を基盤にしている. そして, 学習時に他エージェントの政策 (を基に実行される行動) を考慮に入れるため, 学習関数である Q 関数を, 状態 s と行動 a の関数 $Q(s, a)$ から, 状態 s とすべてのエージェントの行動 $a^1, \dots, a^n$ の関数 $Q(s, a^1, \dots, a^n)$ に拡張している. この拡張した Q 関数を用いた場合の学習空間数は,

$$|S| \times |A^1| \times \dots \times |A^n| \quad (4)$$

である. ここで, $|S|, |A^1|, \dots, |A^n|$ は, それぞれ集合 $S, A^1, \dots, A^n$ の要素数を表す. 式 (4) は, Q 関数を $Q(s, a^1, \dots, a^n)$ と拡張することにより, 学習空間がエージェント数の増加に対して指数関数的に増加することを示している. 強化学習において, 学習空間の指数関数的増加は『次元の呪い』と呼ばれ, 学習を大幅に遅らせる原因となる.

本稿では, 他エージェントの政策推定を利用した MARL 法で, エージェントの行動に依存しない学習関数を利用した新たな MARL 法を提案する. 本稿で提案する MARL 法は, TD 学習 [6] を基盤にしたもので, 学習関数として V 関数 $V(s)$ を用いる. ここで, 学習空間の数は, エージェント数に関わらず, $|S|$ である. 提案する MARL 法では, 他エージェントの政策推定に加えて, 状態遷移確率関数 T の推定を行う. そして, 推定した状態遷移確率関数と, 推定した他エージェントの政

策を基に、自分（エージェント k ）がどの行動を実行すれば、未来にどの状態に遷移するかを式 (5) で予測する。

$$\hat{P}^k(s'|s, a^k) = \sum_{a^{o_1}} \cdots \sum_{a^{o_{n-1}}} \hat{\pi}_k^{o_1}(a^{o_1}|s) \cdots \hat{\pi}_k^{o_{n-1}}(a^{o_{n-1}}|s) \hat{T}^k(s'|s, a^1, \dots, a^n) \quad (5)$$

ここで、 $\hat{P}^k(s'|s, a^k)$ は、エージェント k の観点から、自分が状態 s で行動 a^k を実行したときに状態が s' へ遷移すると予想される確率を表す。 $\hat{T}^k(s'|s, a^1, \dots, a^n)$ は、エージェント k が推定した状態遷移確率関数で、状態 s でそれぞれのエージェントが行動 $a^1, \dots, a^n$ を実行したときに状態が s' へ遷移すると予想される確率を表す。 $\hat{\pi}_k^{o_1}, \dots, \hat{\pi}_k^{o_{n-1}}$ は、エージェント k が推定した他エージェントの政策を表す。ここで、 $o_1, \dots, o_{n-1}$ は、 k 以外のエージェント（他エージェント）のいずれか 1 体に対応するものとする。 $\hat{\pi}^o(a^o|s)$ という表記は、他エージェント o が状態 s で行動 a^o を実行すると予想される確率を表す。本稿で提案する MARL 法では、式 (5) の $\hat{P}^k$ で予測した未来の環境状態 s' に対する V 関数値 $V^k(s')$ を考慮しながら行動選択を行い、強化学習を進行する。

以下で、本研究で採用する他エージェントの政策の推定法、状態遷移確率関数の推定法、そして、提案する MARL 法を示す。

3.1 状態遷移確率関数の推定法

状態遷移確率関数 T の推定には、式 (6) を利用する。

$$\hat{T}^k(s^*|s, a^1, \dots, a^n) = \frac{C(s, a^1, \dots, a^n, s^*)}{N(s, a^1, \dots, a^n)} \quad (6)$$

ここで、 $N(s, a^1, \dots, a^n)$ は、状態 s でそれぞれのエージェントが行動 $a^1, \dots, a^n$ を実行した回数、 $C(s, a^1, \dots, a^n, s^*)$ は、状態 s でそれぞれのエージェントが行動 $a^1, \dots, a^n$ を実行したときに状態が s^* へ遷移した回数を表す。

3.2 他エージェントの政策の推定法

他エージェントの政策の推定には、我々が以前提案した手法 [4] を採用する。以下に、その手法を示す。

時刻 t において、他エージェント o が状態 s_t で行動 a_t^o を実行したとする。そのとき、状態 $s = s_t$ で実行可能な、すべての行動 $a^o \in A^o$ に対して、式 (7) に従っ

て $\hat{\pi}_k^o$ を更新する。

$$\hat{\pi}_k^o(a^o|s) \leftarrow (1 - \theta) \hat{\pi}_k^o(a^o|s) + \begin{cases} \theta & (a^o = a_t^o) \\ 0 & (\text{otherwise}) \end{cases} \quad (7)$$

ここで、 $\theta \in [0, 1]$ は観測した行動を将来の行動予測時にどれくらい考慮するかを決定するパラメータである。式 (7) の更新則によって $\sum_{a^o \in A^o} \hat{\pi}_k^o(a^o|s) = 1$ が保たれることに注意する。本研究では、それぞれのエージェントが他エージェントの行動集合 A^o について予め知っていることを仮定する。式 (7) の推定法が強化学習エージェントの政策の推定に適していることが実験的に示されている [4]。

3.3 未来状態の予測を利用したマルチエージェント強化学習法

以下に、本稿で提案する未来状態の予測を利用した MARL 法の学習の流れを示す。

1. 現在（時刻 t とする）の状態 $s_t \in S$ において、エージェント k は、式 (8) の soft-max 関数で与えられる政策 π^k に従って行動 a^k を選択する。

$$\pi^k(a^k|s_t) = \frac{e^{J(s_t, a^k)/\tau}}{\sum_{b \in A^k} e^{J(s_t, b)/\tau}} \quad (8)$$

ここで、政策 $\pi^k(a^k|s)$ は状態 s で行動 a^k を選択する確率を表す。 τ は温度パラメータと呼ばれ、行動選択のランダムさを調整するパラメータである。 $J(s, a^k)$ は、 V 関数 V^k の $\hat{P}^k$ (式 (5)) に関する期待値で、式 (9) で与えられる。

$$J(s, a^k) = \sum_{s^*} \hat{P}^k(s^*|s, a^k) V^k(s^*) \quad (9)$$

2. エージェント k は、手続き 1 で選択した行動 a_t^k を実行する（ここで、他エージェントも同期して行動 $a_t^{o_1}, \dots, a_t^{o_{n-1}}$ を実行する。環境の状態は、状態遷移確率関数 T に従って s_t から s_{t+1} へ遷移する）。エージェント k は、他エージェントの行動 $a_t^{o_1}, \dots, a_t^{o_{n-1}}$ を観測する。また、環境から直接報酬 r_{t+1}^k を受け取る。

3. エージェント k は、関数 $\hat{\pi}_k^o$ ($o = o_1, \dots, o_{n-1}$) を式 (7) に従って更新し、関数 N 、関数 C をそれぞれ式 (10)、式 (11) に従って更新する。

$$N(s_t, a_t^1, \dots, a_t^n) \leftarrow N(s_t, a_t^1, \dots, a_t^n) + 1 \quad (10)$$

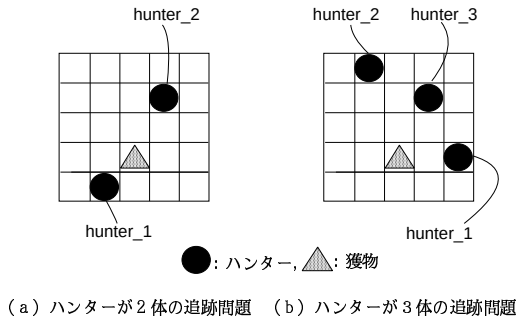


図 1 追跡問題のグリッド空間

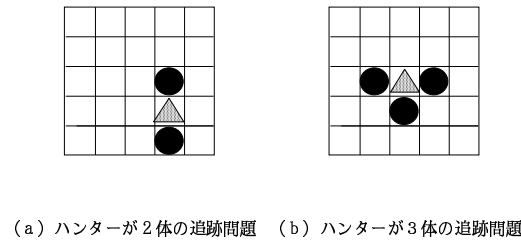


図 2 捕獲状態の例

$$C(s_t, a_t^1, \dots, a_t^n, s_{t+1}) \leftarrow C(s_t, a_t^1, \dots, a_t^n, s_{t+1}) + 1 \quad (11)$$

そして、状態 s_t における V 関数値を式 (12) に従って更新する。

$$V^k(s_t) \leftarrow (1 - \alpha)V^k(s_t) + \alpha(r_{t+1}^k + \gamma V^k(s_{t+1})) \quad (12)$$

ここで、 $\alpha \in (0, 1]$ は学習率と呼ばれるパラメータである (式 (12) の更新式は TD 学習 [6] における更新式と同じものである)。

4. 学習の終了条件を満たしていれば学習終了。そうでなければ t に 1 を加えて、手続き 1 に戻る。

4 実験

4.1 追跡問題

本研究では、実験に使用するタスクとして追跡問題 [8] を取り上げる。追跡問題は、複数のハンターが獲物を追いかけて捕獲する課題である。以下に、本研究における追跡問題の問題設定を示す。

- 2次元 (5×5) のグリッド空間中に、複数のハンターと 1 体の獲物が存在する (図 1)。ここで、グ

リッド空間の上と下、左と右の境界は繋がっているものとする。

- 本研究では、ハンターを『エージェント』と定義する。本研究では、ハンターが 2 体の場合 (図 1 (a)) と、3 体の場合 (図 1 (b)) の 2 つの実験を行う。
- 各時間ステップ毎に、ハンターと獲物は、それぞれ 1 つの行動を同期して実行する。ここで、ハンターが実行可能な行動は、隣接する上、下、左、右のグリッドへ移動する、現在位置に留まる、の 5 通りとする。また、獲物が実行可能な行動は、隣接する上、右のグリッドへ移動する、現在位置に留まる、の 3 通りとする。
- ハンターの目標は獲物を捕獲することである。ここで、捕獲の定義は、
 - ハンターが 2 体の実験では、『2 体のハンターが獲物を上下、あるいは左右から挟んだ状態』 (図 2 (a))
 - ハンターが 3 体の実験では、『獲物に隣接する 4 近傍のグリッドうち、いずれか 3 つのグリッドを 3 体のハンターが占有している状態』 (図 2 (b))

- とする。
- 初期配置から獲物が捕獲されるまでを『1 エピソード』とする。獲物が捕獲されると、ハンターと獲物はグリッド空間中にランダムに初期配置され、新たなエピソードを開始する。

- 環境状態は、それぞれのハンターと獲物の相対位置の組合せ $s = (p^1, \dots, p^n)$ とする。ここで、 p^i ($i = 1, \dots, n$) は hunter- i と獲物の相対位置を表す。例えば、図 1 (a) では $s = ([1, 1], [-1, -2])$ 、図 1 (b) では $s = ([-2, 0], [1, 2], [-1, -2])$ である。
- 獲物は学習を行わず、3 通りの行動の中から 1 つの行動を確率的に選択する。実験では、右へ移動する確率と現在位置に留まる確率をそれぞれ $\frac{2}{5}$ 、上へ移動する確率を $\frac{1}{5}$ としている。この行動選択確率は時不変とする。

以上の問題設定では、報酬関数について言及していないが、報酬関数が式 (2) を満たすように設定された場合、この追跡問題はマルコフゲームの条件を満たす。

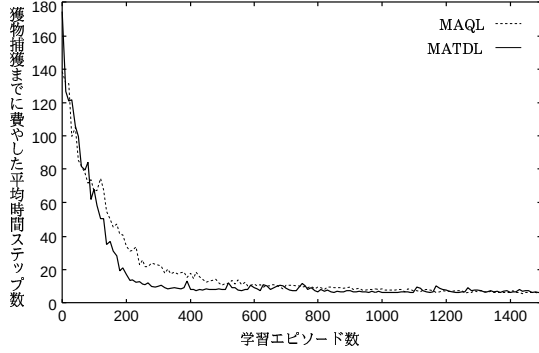


図3 獲物捕獲までに費やした平均時間ステップ数：
ハンターが2体の実験.

4.2 他エージェントの政策推定を利用したマルチエージェント Q 学習法

提案した MARL 法の性能を評価するための比較対象として、本研究では、他エージェントの政策推定を利用したマルチエージェント Q 学習法 [4] (以下 MAQL と書く) を取り上げる。MAQL では、学習関数に Q 関数 $Q(s, a^1, \dots, a^n)$ を用いており、3 章で述べたように、学習空間数 ($|S| \times |A^1| \times \dots \times |A^n|$) は、エージェント数の増加に対して指数関数的に増加する。MAQL の学習の流れを以下に示す。

1. 現在 (時刻 t とする) の状態 $s_t \in S$ において、エージェント k は、式 (13) で与えられる政策 π^k に従って行動 a^k を選択する。

$$\pi^k(s_t, a^k) = \frac{e^{\bar{Q}(s_t, a^k)/\tau}}{\sum_{b \in A^k} e^{\bar{Q}(s_t, b)/\tau}} \quad (13)$$

ここで、 $\bar{Q}(s, a^k)$ は式 (14) で与えられる関数である。

$$\bar{Q}(s, a^k) = \sum_{a^{o_1}} \dots \sum_{a^{o_{n-1}}} \hat{\pi}_k^{o_1}(a^{o_1}|s) \dots \hat{\pi}_k^{o_{n-1}}(a^{o_{n-1}}|s) Q^k(s, a^1, \dots, a^n) \quad (14)$$

2. エージェント k は、手続き 1 で選択した行動 a_t^k を実行する (ここで、他エージェントも同期して行動 $a_t^{o_1}, \dots, a_t^{o_{n-1}}$ を実行する。環境の状態は、状態遷移確率関数 T に従って s_t から s_{t+1} へ遷移する)。エージェント k は、他エージェントの行動 $a_t^{o_1}, \dots, a_t^{o_{n-1}}$ を観測する。また、環境から直接報酬 r_{t+1}^k を受け取る。
3. エージェント k は、他エージェントの政策 $\hat{\pi}_k^o$

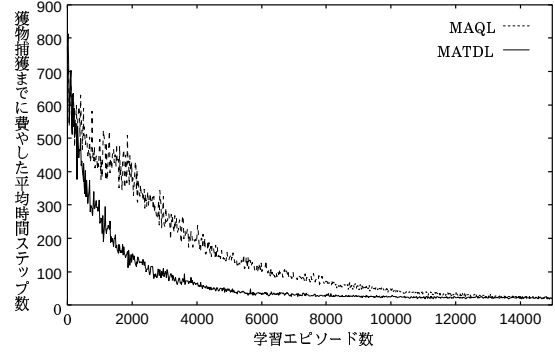


図4 獲物捕獲までに費やした平均時間ステップ数：
ハンターが3体の実験.

($o = o_1, \dots, o_{n-1}$) を式 (7) に従って推定 (更新) する。そして、状態 s_t 、行動 $a_t^1, \dots, a_t^n$ における Q 関数値を式 (15) に従って更新する。

$$Q^k(s_t, a_t^1, \dots, a_t^n) \leftarrow (1 - \alpha) Q^k(s_t, a_t^1, \dots, a_t^n) + \alpha (r_{t+1}^k + \gamma \max_{a^k} \bar{Q}(s_{t+1}, a^k)) \quad (15)$$

4. 学習の終了条件を満たしていれば学習終了。そうでなければ t に 1 を加えて、手続き 1 に戻る。

4.3 実験結果

提案したマルチエージェント強化学習法 (以下、MATDL と書く) と MAQL を追跡問題に適用した。ハンターが2体の場合の実験結果を図3に、3体の場合の実験結果を図4に示す。図の横軸は学習エピソード数、縦軸は1エピソード中で獲物捕獲までに費やした平均時間ステップ数を表す。図の結果は、10学習エピソード毎に、そのときまでの学習性能を評価するため、初期配置を変えた100評価エピソード (このエピソードでは学習を行わない) の実験を行ない、その平均時間ステップ数を示したものである。2つの学習法 (MATDL と MAQL) で使用した学習パラメータは $\alpha = 0.3 \times \text{decay}^{\text{num_ep}}$, $\gamma = 0.9$, $\tau = 0.1 \times \text{decay}^{\text{num_ep}}$, $\theta = 0.5 \times \text{decay}^{\text{num_ep}}$ である。ここで、 decay は減衰係数、 num_ep は学習エピソード数を表す。減衰係数 decay は、ハンター数が2体の実験では $\text{decay} = 0.9977$, 3体の実験では $\text{decay} = 0.99977$ としている。減衰係数 0.9977, 0.99977 は、それぞれ $0.9977^{1000} \approx 0.1$, $0.99977^{10000} \approx 0.1$ となるように選ばれた値である。エージェントが環境から受け

取る直接報酬 r^k は、獲物捕獲時に $r^k = 1.0$ 、それ以外の時に $r^k = -0.05$ としている。すべての $s \in S$, $s^* \in S$, $a^1 \in A^1, \dots, a^n \in A^n$ に対して、V 関数、Q 関数、関数 N 、関数 C のそれぞれの初期値は、 $V^k(s) = 0.0$, $Q^k(s, a^1, \dots, a^n) = 0.0$, $N(s, a^1, \dots, a^n) = 0$, $C(s, a^1, \dots, a^n, s^*) = 0$ としている。また、すべての $s \in S$, $a^o \in A^o$ ($o = o_1, \dots, o_{n-1}$) に対して、関数 π_k^o の初期値は $\pi_k^o(s, a^o) = 0.2$ としている。

図 3, 図 4 の結果は、ハンター数が 2 体の場合、3 体の場合の両方で、MATDL が MAQL よりも学習が速いことを示している。また、獲物捕獲までに費やした平均時間ステップ数の差 (MATDL-MAQL) と比率 (MATDL:MAQL) は、ハンター数が 3 体の場合の方が、2 体の場合より大きいことを示している。

4.4 考察と今後の課題

2つの学習法 MATDL と MAQL の主な相違点は以下の3つである。

1. 学習関数として、MATDL では V 関数 $V(s)$ 、MAQL では Q 関数 $Q(s, a^1, \dots, a^n)$ を用いている。
2. MAQL の学習空間は、ハンター数が 2 体の場合、MATDL の 25 (5^2) 倍、3 体の場合、MATDL の 125 (5^3) 倍である。
3. MATDL では状態遷移確率関数 T を式 (6) により陽に推定しているのに対して、MAQL では学習 (試行錯誤の経験) を通じて、Q 関数中で陰に推定される。

MATDL が MAQL よりも学習が高速である原因は、これら 3 つの違いが影響していると考えられるが、獲物捕獲までに費やした平均時間ステップ数の比率 (MATDL:MAQL) が、ハンター数が 2 体の場合より 3 体の場合の方が大きいことより、項目 2 の学習空間の増加が大きく寄与していることが予想される。より詳細な調査は今後の課題である。

5 あとがき

本研究では、未来状態の予測を利用したマルチエージェント強化学習法を提案した。提案したマルチエ

ージェント強化学習法では、他エージェントの政策と、環境の状態遷移確率関数を推定し、推定したそれらの関数を利用して、どの行動を実行すればどの環境状態に遷移するかを予測した。そして、その予測した環境状態における V 関数値を基にどの行動を実行すればよいかを決定し、強化学習を進行した。提案したマルチエージェント強化学習法では学習関数として V 関数を採用した。提案したマルチエージェント強化学習法をマルコフゲームの枠組みでモデル化した追跡問題に適用し、実験を行った結果、ハンターが 2 体の場合、3 体の場合の両方の実験で、他エージェントの政策推定を利用したマルチエージェント Q 学習法 [4] より学習が高速になった。また、その学習の高速化は、2 体エージェント問題よりも、3 体エージェント問題の方がより顕著に現れた。学習が高速になった原因の詳細な調査は今後の課題である。

参考文献

- [1] R. S. Sutton, and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, Massachusetts, 1998.
- [2] M. L. Littman, "Markov games as framework for multi-agent reinforcement learning," Proc. 11th International Conference on Machine Learning, pp.157-163, New Brunswick, New Jersey, USA, July 1994.
- [3] J. Hu, and M. P. Wellman, "Multiagent reinforcement learning: theoretical framework and an algorithm," Proc. 15th International Conference on Machine Learning, pp.242-250, Madison, Wisconsin USA, July 1998.
- [4] Y. Nagayuki, S. Ishii, and K. Doya, "Multi-agent reinforcement learning: An approach based on the other agent's internal model," Proc. 4th International Conference on Multi-Agent Systems, pp.215-221, Boston, Massachusetts, USA, July 2000.
- [5] C. J. C. H. Watkins, and P. Dayan, "Technical Note Q-Learning," Machine Learning, vol.8, no.3, pp.279-292, 1992.
- [6] R. S. Sutton, "Learning to predict by the methods of temporal differences," Machine Learning, vol.3, pp.9-44, 1988.
- [7] G. Owen, Game Theory: Third edition, Academic Press, San Diego, California, 1995.
- [8] M. Benda, V. Jagannathan, and R. Dodiawalla. "On optimal cooperation of knowledge sources". Technical Report BCS-G2010-28, Boeing AI Center, 1985.

環境状況の変化に応じて 自己の報酬を操作する学習エージェントの構築

森山 甲一 沼尾 正行

Abstract

The authors aim at constructing an autonomous agent learning appropriate actions in several situations of a Multi-Agent environment. Under this aim, the authors constructed an agent having abilities that can handle the agent's own rewards and distinguish the situations, and we saw that the agent can act well in each of the situations of the environment composed of homogeneous agents. In this paper, on the other hand, we check whether the agent can also act well in the environment in which the situations switch over.

1 はじめに

自律エージェントの学習とは可能な選択肢の選択基準を獲得することであり、通常は何らかの外部評価に基づいてその評価値を最大化する選択肢を選ぶようにする。しかし、例えば複数の評価基準が存在してそれらの並立が難しい環境ではどの評価基準に従うかの選択の問題があり、またエージェントが複数存在するマルチエージェント環境では社会的ジレンマの問題がある。

本研究では、マルチエージェント環境における社会

的ジレンマの問題に着目する。社会的ジレンマの下では、個々が利益を追求すればするほど（合理的に行動するほど）実際に得られる利益が減少するため、個々は非合理的に行動することが求められる。社会科学の分野では[1]、この場合に何らかの権威者（政府など）が法律や税制などを整備することによりジレンマ状況を解決する研究がなされている。しかし、例えば我々人間の場合は状況に応じて非合理的に行動できるため、法律などによる細かな規制がない場合でも社会生活を成立させることが可能である。

そこから、権威者による上位の制御がない場合でもジレンマ状況を解決できる人間のような非合理的エージェントを構築することが考えられる。しかし、一方では従来の評価を最大化する合理的行動の学習に関する研究の蓄積があり、それらを有効利用することが望ましい。そこで本研究では、自分に与えられた外部評価を操作して内部評価を生成し、それを従来の学習手法に適用することにより非合理性を実現するエージェントを構築することを目的とする。

確かに、例えば自己の内部評価が他者の利益を含むように予め設計者が設計すればエージェントはジレンマ状況を克服できるだろう。しかし、我々の人間社会のような現実のマルチエージェント環境は常にジレンマ状況であるとは限らず、むしろ動的に変化する。そのため、時と場合によっては他者への影響が無視できる状況も存在し、その中では上記のように設計されたエージェントは明らかに非効率なものとなる。また、そのようなエージェントは開かれたマルチエージェント環境に起こり得る状況の変化には対応できない。つ

Construction of a Learning Agent Handling Its Rewards According to Changes of Environmental Situations.

Koichi Moriyama, 東京工業大学 情報理工学研究所 計算工学専攻, Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology. koichi@nm.cs.titech.ac.jp

Masayuki Numao, 同上, ditto. numao@cs.titech.ac.jp

まり、現実の開かれたマルチエージェント環境で適切に行動するエージェントを構築する場合には、予め設計者が設計するのではなく、エージェント自身がその環境中でジレンマ状況を克服することが求められる。

これらの目的の下で筆者らは、外部評価から内部評価を生成する能力に加えて環境の状況を識別する能力をエージェントに付与し、状況に応じて自己の内部評価を生成するエージェントを構築した[4]。そしてそのエージェントを集めた同質なマルチエージェント環境において適切にジレンマ状況が克服できる一方、非ジレンマ状況においても比較的良好な結果が得られることを示した[4]。しかし、そこでは異なる状況それぞれについて実験を行っており、実験途中で状況が変化する場合を扱っていない。そこで本稿では途中で状況が変化するような実験を行ない、文献[4]で提案した手法について検証する。

本稿は以下のような構成になっている。2 節で本研究で対象とするマルチエージェント環境を分類し、3 節では文献[4]で提案したエージェントの構築手法について述べる。4 節では共有地の悲劇[2][8]をモデル化したゲーム[3]と筆者オリジナルの狭路問題という 2 種類の問題について、途中で状況が変化する実験を行なった結果を述べる。5 節でその実験結果について考察を行ない、6 節でまとめと今後の課題を示す。

2 マルチエージェント環境

本研究では個々のエージェントとエージェント社会全体の関係に着目して、環境を次の 3 つの状況に分類する[4]。

- 非干渉状況: 全てのエージェントが利益を追求する時に社会全体が最良
- 泥沼状況: 全てのエージェントが利益を追求しない時に社会全体が最良
- 競合状況: 一部のエージェントが利益を追求し、残りが利益を追求しない時に社会全体が最良

それぞれの状況で望ましい行動を 2 人ゲームの表の形で 表 1 に示す。

非干渉状況は個々が単に利益を追求すれば良いため非ジレンマ状況に当たる。泥沼状況は個々が自分の利

表 1 各状況における望ましい行動組み合わせ

(a) 非干渉状況		
	利益追求	利益不追求
利益追求	○	
利益不追求		

(b) 泥沼状況		
	利益追求	利益不追求
利益追求		
利益不追求		○

(c) 競合状況		
	利益追求	利益不追求
利益追求		○
利益不追求	○	

益をより良くしようと合理的に行動するほど全体が悪化し個々の利益も減少する状況であり、囚人のジレンマ問題はこれに当てはまる。競合状況は多くの個体が同時に合理的に行動してもそれらの要求を満たせないタイプの状況であり、狭い空間などの排他的な資源によって競合が発生する状況に相当する。

我々の人間社会のような現実の開かれたマルチエージェント環境は、これらの状況が複雑に組み合わさっていると考えられる。従って、そのような環境で行動するエージェントには、上記のそれぞれの状況で適切な学習を行なう能力とともに状況を識別する能力が必要となるだろう。

3 学習エージェントの構築

本研究ではエージェントの学習手法として、明確な教師を必要としない強化学習[6]を用いる。強化学習は報酬というただ 1 つのスカラ強化信号を用いて、それを可能な限り大きくする行動を選択させようとする学習手法である。以下では強化学習のうちの代表的手法である Q-learning[7]を用いる。これは状態 s と行動 a の組み合わせに伴う価値（報酬の見込み値） $Q(s, a)$ （ Q 値）を学習サイクルごとに以下の式により更新し、行動選択の際にこの値を用いるものである。 r_{t+1} は今回の行動提示で得られた報酬を、

α, γ はそれぞれ学習率と割引率を表す^{†1}。

$$\begin{aligned} Q_t(s, a) &= Q_{t-1}(s, a) \quad \text{if } s \neq s_t \text{ or } a \neq a_t, \\ Q_t(s_t, a_t) &= (1 - \alpha)Q_{t-1}(s_t, a_t) \\ &\quad + \alpha(r_{t+1} + \gamma \max_b Q_{t-1}(s_{t+1}, b)). \end{aligned} \quad (1)$$

しかし、これは報酬 r_{t+1} を最大化する合理的行動を選択するように学習する手法である。そのため、マルチエージェント環境、特に泥沼状況や競合状況では各エージェントはどのように学習を行えば良いだろうか。

その 1 つの解として筆者らは学習の際に報酬値を調整することを提案した [4]。そこで用いた手法は、エージェント A_i の時刻 t の報酬 $r_{i,t+1}$ にパラメータ $\lambda_{i,t+1}$ を加えたものを A_i の学習用報酬 $r'_{i,t+1}$ とし、これを (1) 式の r_{t+1} の代わりに用いるものである。なお、以下では表記の繁雑さを避けるために自分 (A_i) を表す添字 i を省略する。

$$r'_{t+1} = r_{t+1} + \lambda_{t+1}. \quad (2)$$

本研究ではまず泥沼状況に有効な λ_{t+1} として

$$\lambda_{t+1} = \sum_{A_k \in N_i \setminus A_i} r_{k,t+1} \quad (3)$$

を用いる。この式で $N_i \setminus A_i$ はエージェント A_i の近隣のエージェントからなる集合 N_i のうち自分自身を除いたものを表している。(2) 式と (3) 式を合わせると、これは自分及び近隣他者の現在の報酬の和を学習用報酬として用いることを意味し、自分が合理的に行動することにより周囲に悪影響が現れる環境において有効である。

また、競合状況に有効な λ_{t+1} として

$$\lambda_{t+1} = r_{t+1} - r_t \quad (4)$$

を用いる。これは報酬の時間変化を考慮したもので、前回の行動による報酬と今回のその差を強調するものである。

これらの λ_{t+1} はそれぞれ対応する環境については有効であるが、異なる環境ではあまり芳しくない。例えば (3) 式を用いた場合、非干渉状況では通常の強化学習のみを用いた場合よりも結果が悪い。また (4) 式を用いた場合には、泥沼状況では文字通り泥沼に

はまってしまふ。ところが非干渉状況でこの式を用いるとエージェントの合理性をより高める結果となり、良好な結果が得られる^{†2}。これらの点から、現実の開かれたマルチエージェント環境で行動するエージェントは λ_{t+1} を適切に選択しなければならない。すなわち、環境が泥沼状況の際には (3) 式を、非干渉状況・競合状況の際には (4) 式を自動的に選択するような能力をエージェントは保持しなくてはならない。

そこで筆者らはこの両者を状況によって自動的に選択する手法を考案した [4]。それには、個々のエージェントがどのようにして状況を識別するかということが重要になるわけだが、提案手法では以下 2 つの状況識別仮定を置き、少なくとも 1 つが満たされる場合に泥沼状況と認識するようにした。

$$Q_{t-1}(s_t, a) < 0 \quad \text{for all } a. \quad (5)$$

$$r_{t+1} < Q_{t-1}(s_t, a_t) - \gamma \max_b Q_{t-1}(s_{t+1}, b). \quad (6)$$

(5) 式は現在どのような行動を実行しても今後正の報酬が見込めないことを意味している。(6) 式は (1) 式から考案したものであり、左辺は実際に獲得した報酬を表している。一方で通常の Q-learning の場合、Q 値が収束した際にはこの両辺が一致することから、右辺が現在までに学習した報酬の予測値を意味していると考えられる。すなわち、(6) 式は実際に得られた報酬が予測値よりも小さいことを表しており、これは報酬の予測値を学習した時点よりも現在の環境が悪化していることを意味している。

ところが、本研究では (2) 式により調整した学習用報酬 r'_{t+1} を Q 値の学習に用いているため、(6) 式の左辺は学習用報酬 r'_{t+1} でなくてはならない。しかし、これは r'_{t+1} を計算するためにどちらの λ_{t+1} を選択するかを決めるための式であるので、 r'_{t+1} をここで用いることは出来ない。そのため実際の獲得報酬 r_{t+1} で代用していることに注意を要する。

筆者らは、以上の仮定を満たす場合に (3) 式を、満たさない場合に (4) 式を用いて報酬を操作することを提案している [4]。

^{†1} 以下、時刻の表記は Sutton らの流儀に依る [6]。ここでは時刻 t の状態 s_t で行動 a_t を実行すると、状態 s_{t+1} に遷移し報酬 r_{t+1} が与えられるとする。

^{†2} 非干渉状況では合理的行動が望ましいことに注意。

4 実験

本研究では実験環境として、共有地の悲劇 (n 人 囚人のジレンマ) [2][8] をモデル化したゲーム [3] と筆者オリジナルの狭路問題を用いる。いずれの実験も同質なエージェント 10 台で行なう。実験に用いるエージェントは、行動選択をランダムに行なうランダムエージェント、通常の強化学習を行なう通常エージェント、(3) 式を用いて報酬を操作する近隣報酬合計エージェント、(4) 式を用いる報酬差分エージェント、更に (3) 式と (4) 式を (5)(6) 式を用いて自動的に選択する自動選択エージェントの 5 種である。学習率 α ・割引率 γ はともに 0.5 とする。行動選択には Q 値から求めた温度 $T = 1$ のボルツマン分布

$$p_t(a) = \frac{\exp(Q_{t-1}(s_t, a)/T)}{\sum_b \exp(Q_{t-1}(s_t, b)/T)} \quad (7)$$

による確率 $p_t(a)$ を用いる [6]。

4.1 共有地の悲劇ゲーム

これは共有地の悲劇 (n 人 囚人のジレンマ) [2][8] をモデル化したゲーム [3] である。以下ではこのゲーム環境において泥沼状況と非干渉状況を作り出し、エージェントの集合がどのように振る舞うかを確認する。

各エージェントは利己的・協力的・利他的の 3 種の行動のいずれかを一斉に提示し、エージェント社会全体における行動の組み合わせにより決められる報酬を獲得する。この一連の流れを 1 サイクルと称する。エージェント A_i の行動に対して社会全体に対するコスト c_i が付随する。ここでは利己的・協力的・利他的な行動についてそれぞれ $+\Delta r_c, 0, -\Delta r_c$ の各値をとるものとする。泥沼状況においては $\Delta r_c = 1$ 、非干渉状況においては $\Delta r_c = 0$ とした。一方、エージェント A_i の報酬はそれぞれ $3 - r_c, 1 - r_c, -3 - r_c$ とする。ここで r_c は共有コストと称し、 $r_c \triangleq \sum_i c_i$ と定義する。この場合に、泥沼状況では全てのエージェントが利他的に行動した場合に全体の報酬合計が最大 70 となり、全てが利己的に行動した場合に -70 と最小になる。一方で非干渉状況では $r_c \equiv 0$ となり、各エージェントの獲得する報酬は他のエージェントに依

存しないため、全てのエージェントが利己的に行動した場合に報酬合計は最大 30 となる。この環境で重要な点は、個々のエージェントの立場から見ると、泥沼状況においても他者の行動に関係なく利己的行動が常に最大の報酬をもたらすことである。

ここで一例としてゴミの分別を考えてみよう。焼却の際のダイオキシンの発生を防ぐために、可燃ゴミと不燃ゴミを分別するように求められているとする。ダイオキシンは塩素化合物を含むゴミを約 400°C で燃やすことにより生じ、発ガン性が疑われている [5]。この場合に上記の利己的・協力的・利他的の 3 種の行動はそれぞれ「分別しない」「自分のゴミは分別する」「集積場にある他者のゴミも分別する」となる。もし分別しなかった場合には、自治体はダイオキシンの発生を防ぐために税金を用いて人を雇い、ゴミの分別をさせることになるだろう。この場合に使われる税金が上記の共有コストに相当する。一方では、たとえ利他的に集積場の他者のゴミまで分別するとしても、その人が自治体に支払う税金が減るわけではない。そうするとゴミの分別は面倒なので、分別をしないという利己的な行動が最も好ましいと思う人が生じて不思議ではない。しかし、もし住民全員が利己的な行動を採ったらどうなるだろうか。これは増税をもたらすかもしれないため、泥沼状況となるだろう。

しかし、ダイオキシンは約 400°C で燃やすことにより生じるため、これよりも高い温度でゴミを焼却するようにすれば発生を防ぐことが可能である。また、ダイオキシンを除去する装置も世の中に存在する [5]。従って、高温でゴミを焼却するダイオキシン除去装置付きの焼却炉を自治体が建設した場合には、それ以後ダイオキシンの観点からは住民は可燃ゴミと不燃ゴミを分別する必要がなくなる。そうすると分別をしないことによる共有コストの増加は 0 となり、非干渉状況に移行したと考えられるだろう。

(3) 式に現れる近隣エージェント集合 N_i は

$$N_i \triangleq \{A_k \mid k = (i+j) \bmod 10, j = 0, 1, 2, 3\} \quad (8)$$

で定義する [3]。更に、各エージェントの保持する状態は近隣エージェントの行動の組み合わせとする。

最初は泥沼状況だが 30000 サイクル後に突然非干渉状況へ変化した場合の実験結果を図 1 に示す。こ

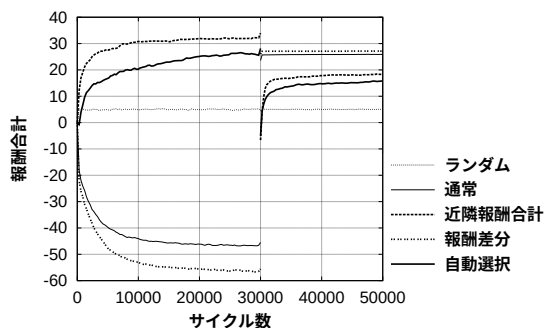


図1 共有地の悲劇ゲームの実験結果

のグラフは100回の実験の平均をベジエ曲線で近似したものであり、状況が変化する30000サイクルでグラフを分割している。縦軸が10台のエージェントの報酬合計、横軸がサイクル数である。30000サイクルまでを見てみると「通常」は時間とともに報酬和が減少している。これは個々が合理的行動をするほど社会が悪化して個々の利益も減少する泥沼状況が生じていることを意味する。また、「近隣報酬合計」が最良の結果をもたらす一方で、「報酬差分」は「通常」よりも悪い結果となっている。「自動選択」は「近隣報酬合計」よりは遅れるが上昇している。

また、最初から非干渉状況である場合の実験結果を図2に示す。グラフによると「報酬差分」が最良の結果をもたらしており、それに続くのが「通常」となっている。一方では、「近隣報酬合計」が途中で上昇が止まってしまっており、「通常」より悪くなっている。「自動選択」は「近隣報酬合計」と「報酬差分」のほぼ中間の結果となっている。

ここで、図1の30000サイクル以後と図2を比較すると、状況が途中で変化する場合の結果が初めから非干渉状況であるものよりも悪くなっていることが分かる。これは「ランダム」を除く全てのエージェント集合に言えることであるが、(2)式により学習用の報酬を調整する3種のエージェントのうち「自動選択」の悪化が大きいことが見て取れる。

4.2 狭路問題

以下では筆者オリジナルの狭路問題を用いて競合状況と非干渉状況を作り出し、エージェント集合がど

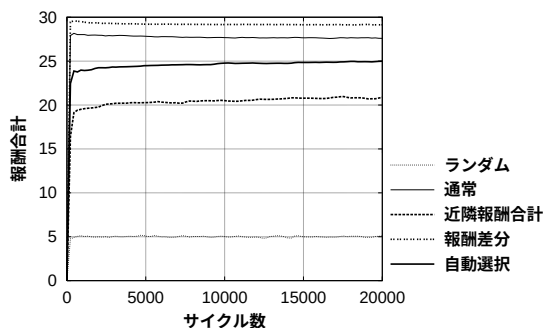


図2 共有地の悲劇ゲーム（非干渉状況）の実験結果

のように振る舞うかを確かめる。

狭路問題では、路上駐車のため道路に車がすれ違うことのできない狭い部分があり、狭い部分の両端に同時に2台の車が現れたと仮定する(図3(a))。問題は、この狭い部分を2台の車がいかにして早く通過するかというものである。車(エージェント)の取り得る選択肢として「進む」と「待つ」の2つがあり、各エージェントにとって最良の結果は相手を待たせて自分が先に通過することである。しかし両者が「進む」を選択すると、狭い部分ですれ違うことが不可能なので結局両者とも通過できない。一方で両者が共に「待つ」を選択すると何も状況は変化しない。そのため、この場合にはエージェントの一部が利益を追求(「進む」を選択)し、残りが利益を追求しない(「待つ」を選択)ことが求められ、競合状況となる。

もし、ある時点でこの路上の車が両方とも移動してしまったとすると(図3(b))、もはや2台の車は相手の車を意識することなくこの場所を通過すること

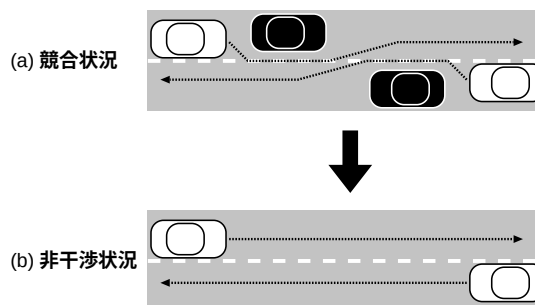


図3 狭路問題

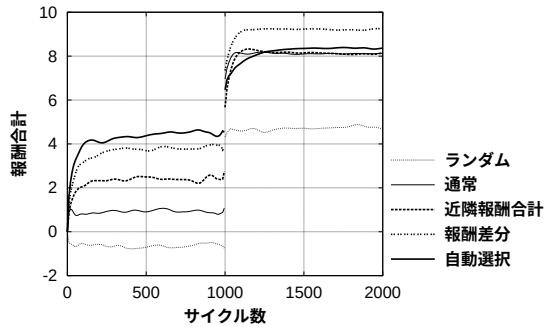


図 4 狭路問題の実験結果

が出来るようになる。従って、この場合には非干渉状況となる。

実験では 10 台のエージェントをランダムに 2 台ずつに分け、各組ごとに両者が行動を提示する。競合状況では各組で 1 台のみが「進む」となった場合にその「進む」を提示したエージェントは通過成功となる。一方で「待つ」を提示したエージェントは（相手なしで）再び行動を提示する。また、両者の行動が一致した場合には両者とも再び行動提示を行なう。毎回の行動提示の際に両者には負の報酬が与えられ、「進む」を提示して通過に成功したエージェントには正の報酬が与えられる。非干渉状況では「進む」を提示したエージェントは無条件で通過に成功し正の報酬が与えられ、「待つ」を提示したエージェントには負の報酬が与えられて再び行動提示を行なうことになる。どちらの状況においても $2^2 = 4$ 回の行動提示が行なわれた後にはエージェントが残っていてもその組は終了とする。その場合には正の報酬は与えられない。このようにして全ての組が終了したならばそれを 1 サイクルと称し、再び最初から繰り返す。

エージェントに与えられる報酬は、「進む」を提示して通過に成功したエージェントに +1、それ以外には行動提示ごとに -0.5 とした。つまり報酬合計の理論上の最大値は競合状況で $(1 + 0.5) \times 5 = 7.5$ 、非干渉状況で $(1 + 1) \times 5 = 10$ となる。各エージェントは行動提示ごとに学習を行なう。

近隣エージェント集合 N_i は組分け後の自分と相手とし、もし相手が先に通過に成功した場合には自分のみとする。各エージェントの状態は、自分の組の残り

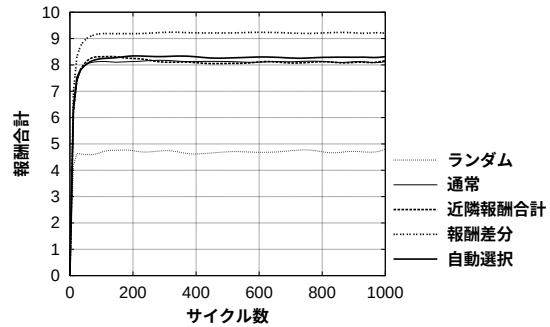


図 5 狭路問題（非干渉状況）の実験結果

エージェント数と相手 ID の組み合わせとした。

最初は競合状況であったが 1000 サイクル後に非干渉状況となった場合の実験結果を図 4 に示す。描画方法は図 1 と同様であり、状況が変化する 1000 サイクルでグラフを分割している。1000 サイクルまでのグラフによると「自動選択」が最良の結果をもたらしており、それに続くのが「報酬差分」となっている。このように、「自動選択」がその元となっている「報酬差分」よりも良い結果となった原因の究明は今後の課題である。「近隣報酬合計」も「通常」よりは良いのだが「報酬差分」より悪いことが分かる。

また共有地の悲劇ゲームと同様に、初めから路上駐車のない非干渉状況における実験結果を図 5 に示し、これを図 4 の 1000 サイクル以後と比較する。この場合にはどちらも結果に差はなく、「報酬差分」が最良で「ランダム」を除く他のエージェント集合はほぼ同様の結果となっている。

5 考察

本稿では文献 [4] で筆者らが提案した手法について、途中で状況が変化する環境で実験を行なった。この実験結果について以下で考察を行なう。

まず 4.1 節の共有地の悲劇ゲームでは、「ランダム」を除く全てのエージェントについて、環境が泥沼状況から非干渉状況に移行した後（図 1 の 30000 サイクル以後）の結果が初めから非干渉状況であった実験の結果（図 2）より明らかに悪くなっている。これは状況変化前に学習した結果の影響が出ているためと考えられる。

また、報酬を調整した 3 種のうち「自動選択」の結果の悪化が最も大きく、「近隣報酬合計」よりも劣るものとなっている。この原因としては、状況識別仮定のうちの (6) 式の左辺に r'_{t+1} の代わりに r_{t+1} を用いたことが影響していると考えられる。このゲームにおける泥沼状況では獲得報酬 r_{t+1} が最小となる利他的行動が望まれるため、必然的に r'_{t+1} に対する λ_{t+1} の影響が大きくなり、 r'_{t+1} と r_{t+1} が全く異なる値になると思われる。すると学習の結果である Q 値も r_{t+1} から得られる値とは異なるものとなり、結果として (6) 式の判定に誤りが生じるのではないだろうか。またより本質的な問題として、(6) 式では学習が収束した状態を想定しているが、実験開始直後ではそれは望めないという点と、そもそも環境の非マルコフ性により Q -learning では収束性すら保証されていないという点が問題となっているのかもしれない。

一方、4.2 節の狭路問題の場合には、競合状況から非干渉状況に移行した後（図 4 の 1000 サイクル以後）の結果は初めから非干渉状況であった実験の結果（図 5）とほぼ同じである。すなわち、この場合には状況変化前の学習の悪影響が見られない。これは (4) 式を用いれば状況変化前後どちらにも対応できるというように競合状況と非干渉状況は性質が似ているためと考えられる。しかし、非干渉状況において「自動選択」が「近隣報酬合計」とほぼ同じ結果になったということは、提案手法による状況の識別がうまく機能していないことを意味している。

もし、エージェントの知覚能力がもっと豊かであったらどうであろうか。共有地の悲劇ゲームの例では、自治体が新焼却炉を建設した場合に通常は住民に対して何らかのアナウンスがあるだろう。また狭路問題の例では、運転者は路上駐車がなくなったことを視覚から認識できるはずである。これらのようにエージェントが豊かな知覚能力により状況変化を認識できるならば、エージェントは状況が変化する 1 つの環境を状況が不変な複数の環境に分割することが可能となる。この場合には、それぞれの環境中では文献 [4] における実験と同等となり、過去の学習の悪影響を受けずに学習することが可能である。あるいは (5)(6) 式を用いて状況を推測しなくても、初めから (3) 式か (4) 式

のどちらを利用すれば良いかが自明となるかもしれない。しかし、この議論は対象とする問題をエージェントの知覚の問題に転嫁しただけであり、状況変化が知覚できない場合については何も述べていないことに注意が必要である^{†3}。本研究はこの点を踏まえて、与えられる外部評価を基に限られた知覚によって学習を行ない、環境に適応するエージェントを構築することを目的としている。

6 まとめ

本稿では筆者らが提案した手法 [4] について新たにを行った実験の結果を示した。文献 [4] ではマルチエージェント環境の 3 種それぞれの状況が不変な場合の実験のみを行っていたが、本稿では途中で状況が変化する場合の実験を行った。その結果、共有地の悲劇ゲームにおける泥沼状況から非干渉状況への移行の際には、提案手法は状況の変化に対応する能力が乏しいことが判明した。従って提案手法は、いろいろな状況が組み合わさっている現実の開かれたマルチエージェント環境に対応するためのエージェントを構築するという観点からは不合格であるといえる。しかし、考察で言及した通り、(6) 式の左辺を r_{t+1} で代用した点を改良することによって、少なくとも報酬を調整する個々の手法 ((3)(4) 式) と同等な適応力を得られるのではないと思われる。これは行動選択用の Q 値と状況識別用の Q 値を分けて考えることで可能になると思われるが、単純に考えても空間使用量が倍になる点がデメリットである。一方で狭路問題の場合には、非干渉状況において提案手法による状況の識別がうまく機能しないことが判明した。この原因を究明してより良い状況識別法を構築することも必要となるだろう。

それ以外の今後の課題としては、状況識別の自動化の追求、異質なエージェントからなるマルチエージェント環境における提案手法の有効性の確認、(3) 式における近隣エージェントからの情報の不要化と異なる強化学習法の適用が考えられる [4]。

^{†3} これは、資源の枯渇の問題などの状況変化が徐々に起こる問題について顕著であると思われる。

References

- [1] 出口. 複雑系としての経済学. 社会科学のフロンティア 6. 日科技連, 東京, 2000.
- [2] G. Hardin. The Tragedy of the Commons. *Science*, 162:1243–1248, 1968.
- [3] S. Mikami and Y. Kakazu. Co-operation of Multiple Agents Through Filtering Payoff. In *Proc. 1st European Workshop on Reinforcement Learning, EWRL-1*, pp. 97–107, Brussels, Belgium, 1994.
- [4] 森山, 沼尾. 自己の報酬を操作する学習エージェントの構築. 人工知能学会 第 45 回人工知能基礎論研究会資料 (SIG-FAI-A101), pp. 15–20, 東京都, 2001.
- [5] T. G. Spiro and W. M. Stigliani. *Chemistry of the Environment*. Prentice-Hall, Englewood Cliffs, NJ, 1996. (岩田 元彦, 竹下 英一 訳. 地球環境の化学. 学会出版センター, 東京, 2000) .
- [6] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [7] C. J. C. H. Watkins and P. Dayan. Technical Note: Q-learning. *Machine Learning*, 8:279–292, 1992.
- [8] X. Yao and P. J. Darwen. An Experimental Study of N-Person Iterated Prisoner's Dilemma Games. *Informatica*, 18:435–450, 1994.

自律学習における CITTA による部分知識の結合

山川 宏^{*1}、宮本 裕司^{*2}、馬場 孝之^{*1}、岡田 浩之^{*1}

本報告では、自律学習というフレームワークにおいて、タスク学習において事前学習で獲得した部分知識をモジュールとして組み合わせて利用するために開発した、マルチモジュール問題解決アーキテクチャである CITTA (Cognition based InTelligent Transaction Architecture) について述べる。自律学習のフレームワークでは、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まってない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う。

事前学習で獲得する部分知識にはセマンティクスが与えられていないので、CITTA は本来同一の意味を持つ情報についてはスタック上に積み上げて扱う。また、マルチモジュール環境では各モジュールが取り扱う変数の制御可能性が動的に変化するので、CITTA はセンサ入力と行動出力を統一的に扱う一般化状態を用いる。

最後には「RWC 2001 Final Exhibition & Symposia」で行ったナビゲーション課題によるデモンストレーション「ドア&鍵課題の自律学習による解法」において、自律学習機能により、事前学習により予め獲得した部分環境知識を、タスク学習で CITTA によって部分環境知識を組み合わせて利用することで、新たな場面への適応が可能となることをお見せする

1 はじめに

我々は、経済産業省の国家プロジェクトであるリアルワールドコンピューティングにおいて、図 1 に示すような、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まってない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う自律学習というフレ

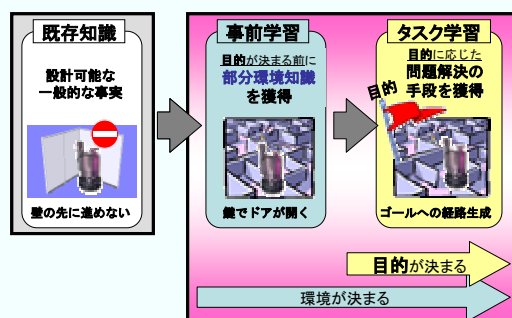


図1 自律学習の枠組みの提唱

ームワーク[1]を提唱してきており、これまでに、状況分解[2]と強化学習を組み合わせたナビゲーション実験[3]などを行っている。

我々は自律学習の実現のために、3つの要素技術を開発した。1つ目に事前学習において、部分環境知識を抽出するための Matchable 状況分解手法[2]で、本技術ではスプレッドシート状に整形されたデータに対して、特徴量とイベントの同時選択により関係の強い複数の部分情報を自動的に抽出する。

2つ目の要素技術は、個々のモジュールにおける認知距離学習による問題解決器である。学習より獲得した任意の状態間の行動によって結び付けられた認知距離を用いて、実行時の計算コストを抑えながらの与えられたゴールに応じた行動決定が可能である[4]。

3つ目の要素技術は、マルチモジュールアーキテクチャとしての CITTA がある。タスク学習では、CITTA を用いて事前学習で獲得した部分知識をモジュールとして組み合わせて利用する。

2 自律学習というフレームワーク[1]

実世界環境からの情報を扱う従来システムでは、人間があらかじめ獲得し、設計した環境に関する知識を利用してタスク学習を行っていたが、その知識獲得に要する設計コストが大きく問題となっていた。なぜなら、システムは環境ごとに知識獲得を行う必要があるためである。

Combining Partial Knowledge using CITTA in Autonomous Learning Framework

Hiroshi Yamakawa, Takayuki Baba, Hiroyuki Okada,
(株)富士通研究所, FUJITSU LABORATORIES LTD
Yuji Miyamoto*2, (株)富士通ビー・エス・シー研究所,
FUJITSU BROAD SOLUTION & CONSULTING Inc.

そこで我々は個別の環境に依存しない普遍的な知識をシステムに設計して与えた後に、環境に関する知識獲得を事前学習として行う、自律学習の枠組みを提唱した。これによれば、環境に関する知識の設計コストが大幅に低減できる。

自律学習という考え方は、人間の認知発達からヒントを得たものです。つまり子供の時には特定の目的を持たない行動を行いながら環境に関する知識を蓄積してゆき、大人になったときには既に蓄えていた知識を利用して目的に応じた効率的に学習やタスクの遂行を行える能力を実現し様としています。そういう意味では、認知発達の過程では学習は多段階で行われており、単純に 2 段階に分けられるものではありません。先行する学習が後の学習を効率的に行えるようにサポートするブーツストラップな関係を作り上げることが重要であると考えています。しかし、行動の目的の有無により事前学習とタスク学習を切り分けは、最もわかりやすいと考えています。

事前学習とタスク学習の具体的な手法としても、様々な選択肢があるとは思いますが、現在我々は環境に関する知識獲得の手法として Matchable 状況分解手法[2]を利用し、これにより得られた部分環境知識を組み合わせ用いれば、タスク学習においては計算コストの削減や、より少ない経験からの汎化能力が実現できます。

3 マルチモジュールアーキテクチャ(CITTA)

CITTA により、事前学習で得られた部分環境知識を組み合わせ利用する。今回提案する CITTA(図2参照)では、今まで我々が提案してきた CITTA[5,6]に対して、他の二つの技術と組み合わせるため、情報交換に利用する共有メモリのスタック化やコントローラの導入などの改良を行った。

3.1 CITTA の構成と協調動作

CITTA はその内部に共有メモリ、モジュール群、コントローラを持ち、全体として環境からの現在の状態を入力し、ゴール解決のための処理を行い、ゴールへ移動するための次の状態を出力する。CITTA は環境と交互に動作し、これを 1 サイクルとする。外部環境およびモジュール間との情報交換には共有メモリが用いられる。

CITTA は複数のモジュールを管理し、ある現在の状態において処理が行なえるモジュールを使用して問題解決を行なう。

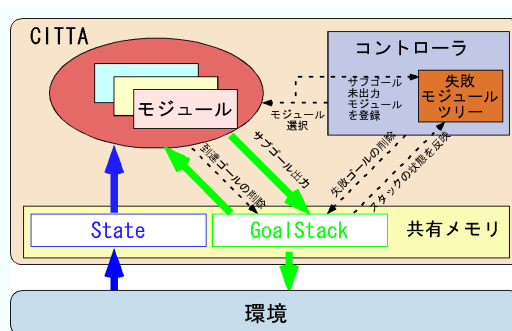


図2 CITTA の基本構成

この図では、現在の状態(State)のベクトルとゴールnスタック(Goal Stack)を分けて書いてあるが、実際にはノードごと現在の状態とゴールのスタックが存在する。

使用するモジュールはコントローラによって選択される。

共有メモリは情報の最小単位であるノードのベクトルで、ノードごとに一つの現在の状態と、実現したいゴールのスタックを持つ。

コントローラではモジュールの制御とスタックの制御を行なう。1 サイクルごとに、実行処理を行う 1 つのモジュールを選択する。一度選択され実行処理を行なったモジュールはサブゴールを出力できる間は連続して選択され、実行処理を行う。サブゴールを出力できない場合に、サブゴールを出力できるモジュールが見つかるまで、ランダムにモジュール選択を行う。

また、スタック上のゴールで、全てのモジュールが実行できない(全モジュールがサブゴール未出力)ゴールはスタックから削除する。

各モジュールは、接続されているノードとの情報の受け渡しとして、現在の状態、ゴールの取得と、サブゴールの出力を行う。処理としては、学習処理、到達ゴールの削除処理、実行処理の3つがある。学習処理、到達ゴールの削除処理は全てのサイクルで行う並列的な処理であるが、実行処理についてはコントローラによって選択されたモジュールのみ行う(逐次処理)。

学習処理は、取得した現在の状態を利用してモジュールごとの学習を行う。到達ゴールの削除処理は、自ら出力したサブゴールに到達していればスタックから削除する。

実行処理は取得した現在の状態とゴールについての問題解決を行い、次の状態をサブゴールとして出力して共有メモリ上のゴールスタックにプッシュする処理を行なう。しかしながら、以下のような3つサブゴール未出力条件を満たす場合には、モジュール内の処理を行う以前に実行動作は中止される。

- 現在のゴールに対して既に失敗している
 - 現在の状態が既にゴール到達している
 - 前サイクルに出力したサブゴールに到達していない
- また、モジュール内の処理をした後に以下のような条件になった場合にもサブゴールの出力を行わない。
- 内部処理でサブゴールを出力できない
 - 前回と同じサブゴールを出力しようとした
 - ゴールと同じサブゴールを出力しようとした

以上のような動作により、スタックの最上部のゴールを監視しているモジュールの中から動作可能なモジュールの一つが動作を行うことにより全体として協調して問題解決が図られる。

3.2 CITTA の特徴

このアーキテクチャは、一種の黒板モデルと考えられるが、事前学習により獲得した部分知識を利用するために以下のような点で特徴がある。

- **セマンティクスラベルが存在しない**

部分知識にはセマンティクスをあらわすラベルが無いので、本来同一の意味を持つ情報を、スタック上に積み上げるなどして、その同一性を保持する。このために、システム内に同じ意味を持つ情報は分散して存在しない。

- **一般化状態の導入**

マルチモジュール環境では各モジュールが取り扱う変数の制御可能性が動的に変化し、それを事前学習の段階で知することは困難なので、CITTA はセンサ入力と行動出力を統一的に扱う一般化状態を用いる。

4 デモンストレーション

図3に示す、「RWC 2001 Final Exhibition & Symposia」で行ったナビゲーション課題によるデモンストレーション「ド

ア&鍵課題の自律学習による解法」により、自律学習の効果を示す。

4.1 課題の基本設定

環境中に置かれたエージェントは以下の情報の入出力を行う

- 位置: 部屋内の 2 次元座標位置
- 行動: 前後左右方向への移動
- 存在物: 現在位置のもの (ドア、鍵置場、電話置場、エサ)
- 所持品: 現在持っている物 (なし、鍵、電話)
- 部屋: 部屋の ID
- ドア条件: ドアが開く所持品を示す ID

これらは、一般化状態であるから特にセンサと行動は区別されていない。

4.2 事前学習

事前学習では、エージェントが 9 個の異なる環境でランダムウォークし、そのデータを用いて Matchable 状況分解します。ここでの状況分解では 2 時刻を一まとめにして扱う時系列状況分解の技術が用いられ、これにより部分マルコフ遷移を取り出すことができます。

状況分解は属性とイベントの同時選択により、よく起こる出来事の部分集団 (部分状況) を抽出します。ここで、各部分状況は内部に規則性に基づいて評価されます。) よって、状況分解による事前学習を行うことで、「鍵とドアの関係」や、「位置と移動の関係」などについての部分環境知識が獲得されます。この例では合計 14 個の部分環境知識 (部分状況) が抽出されました。その結果を図4に示します。この中では、位置、行動、存在物、部屋を含む部分状況を取り出されており、これは移動に関するモジュールとして動作します。また、存在物、所持品、ドア条件を含む部分状況はドアを開けるためのモジュールとして動作します。

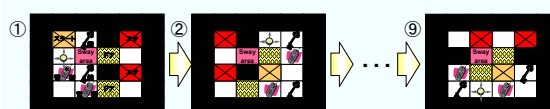


図 3 事前学習を行う環境

部屋ごとにマップが異なり、鍵でドアが開く部屋と電話でドアが開く部屋などがある。

No	位置	行動	存在物	持ち物	部屋	ドア条件
1	場所1	↑	空	空	1	1
2	場所4	→	鍵	鍵	1	1
3	場所3	↓	空	鍵	1	1
4	場所5	↓	ドア	鍵	1	1
5	場所6	↓	空	鍵	1	1
6	場所9	←	エサ	鍵	1	1
...	...	...	...	...	...	...
1000	場所7	↓	空	空	1	1
1001	場所1	↑	空	空	2	1
...	...	...	...	...	...	...
9000	場所1	↑	電話	電話	3	3

図 4 事前学習で抽出される部分状況

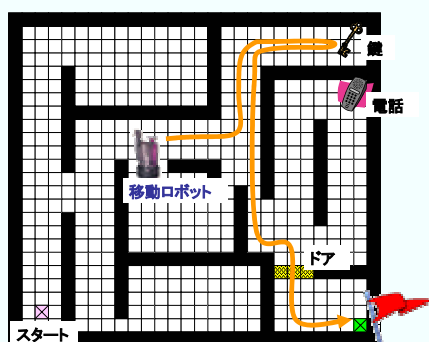


図 5 タスク学習に用いられた部屋の例

4.3 タスク学習

この場合の CITTA の動作は、エサが存在する場所に行くことがゴールとして与えられた場合には、そのゴールを実現可能であると判断した「位置と移動の関係」をもつモジュールがまず動作することがおこる。

部分環境知識を利用すれば、ゴールを与えられた以降のタスク学習での行動決定に必要な経験の量を小さくできる。このため必要に応じて鍵を取りながらゴールへ向かう経路生成を少ない経験で行えるようになった。

また自律学習では、タスク学習の経験も分解して蓄えるので、環境が変化しても部分的な経験を組み合わせることで、新たな環境に素早く対応できる。

5 おわりに

我々が提案している、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まってない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う自律学習というフレームワークについて述べた。

事前学習としての状況分解で獲得した部分知識を、タスク学習においてモジュールとして組み合わせて利用するために、改良したマルチモジュールアーキテクチャである CITTA の提案を行った。ナビゲーション課題において、事前学習により予め獲得した部分環境知識を、タスク学習で CITTA によって部分環境知識を組み合わせることで、新たな場面への適応能力が向上することが示された。

我々の研究の目的は、柔軟な学習能力を実現であり、そのためにシステムの内部構造を学習により獲得することを目指してきた。今回の成果はそのようなことがある程度可能になってきたことをデモンストレーションすることが目的である。そのため、マルチ・エージェント技術の点から見ると、エージェント間の協調や競合解消などに関しては、既存の技術が十分に盛り込まれていない。一方で学習によりシステムの内部構造を獲得しようとする場合には、「セマンティクスラベルを利用できない」、「センサと行動を区別しづらい」などの困難があり、それらに適応しうるアーキテクチャとして CITTA 提案を行っている。この研究は学習システムとしての工学的応用のみならず、人間の脳機能を理解する上でも有用であると考えているので、それらとの関係付けも探ってゆきたい。

なお本研究の一部は RWC 予算により行われた。

参考文献

- [1] N. Watanabe, H. Okada, H. Yamakawa and T. Mohri. (2000). "Situation Decomposition Algorithm for Autonomous Learning," Proc. RWC2000, pp. 219-222.
- [2] 山川, 岡田, 渡部. (1999). "規則性を持つ部分データを抽出するアルゴリズムの提案," Proc. IBIS'99, pp.75-80.
- [3] 伊藤, 岡田, 山川, 大森. (2000). "Matchable 状況分解に基づく Multi-module 強化学習による移動ロボットの経路探索," 信学技報, NC99-99, pp. 47-54.
- [4] 山川, 岡田 馬場. (2000). "認知距離を用いた問題解決器の提案," Proc. MACC2000.
- [5] 山川, 岡田, 渡部, 松尾. (1998). "一般化状態における協調と競合のためのエージェント内部機構," Proc. MACC'98.
- [6] 山川, 岡田, 渡部, 松尾. (1998). "実世界で自律学習を行うための分散知能アーキテクチャ -CITTA-, " 第 12 回人工知能学会全国大会, pp.455-456.

ネット情報を使った意思決定のための知識獲得エージェント

藤本 和則 山本 裕

本稿では、インターネット上の情報から意思決定に必要な知識を自動獲得する機構について、これを複数の知識獲得エージェントの協調により実現する協調獲得モデルを提案する。まず、統計的決定に必要な「代替案の集合、情報の集合、確率分布」の知識をそれぞれ獲得する 3 つの知識獲得エージェントについて述べる。そして、これらのエージェントが、容量の限られた通信路を共有しながら知識を収集／獲得する協調獲得モデルを提案する。

1 はじめに

近年、自然言語処理や知識獲得技術の進歩[6][10]により、インターネット上の情報から様々な知識がある程度の精度で自動獲得できるようになってきた。こうした状況に合わせて、ユーザの意思決定に必要な知識をネット上の情報から自動構成する研究が始められている[1][5]。我々は、特に、不確実な状況下で合理的な意思決定を行うのに必要な知識について、これをユーザとネット情報から自動獲得する意思決定支援の枠組について研究を進めている[8][9]。

我々は、これまでに、ユーザから「状況の集合と効用関数」を、ネット情報から「代替案の集合、情報の集合、確率分布」をそれぞれ獲得する知識獲得システムの構成を示した[9]。これら 5 種類の知識の獲得によ

り、各代替案の期待効用(得られる利益の見込み)を計算でき、その大きな代替案を薦めるという意思決定支援が可能となる。本稿では、これらの知識のうち、インターネットを検索しながら獲得する「代替案の集合、情報の集合、確率分布」に着目し、これらを効率的に収集／獲得する知識獲得機構について述べる。

インターネット上の情報から知識を獲得する状況では、獲得に用いる通信路の容量は限られるので、意思決定の質を上げる効果の大きい知識を優先して獲得する機構の実現が重要となる。本稿では、こうした知識獲得機構の一つとして、「代替案の集合、情報の集合、確率分布」のそれぞれを獲得する知識獲得エージェントの協調に基づく協調獲得モデルを提案する。このモデルに基づいて知識獲得を行うことにより、意思決定に用いる知識ベースをより早く価値の高いものとして構成することが可能となる。まず、2 章で、代替案の集合、情報の集合、確率分布をそれぞれ獲得する知識獲得エージェントについて説明する。3 章では、各知識獲得エージェントが意思決定の質を上げるために、協調しながら知識を収集／獲得する機構について述べる。

2 知識獲得エージェント

図 1 に知識獲得システムの構成を示す。図に示すように、知識獲得システムは、ネット上の情報から「代替案の集合、情報の集合、確率分布」を、ユーザから「状況の集合、効用関数」をそれぞれ収集／獲得する。本章では、これらの知識のうち、インターネット上の情報から獲得する代替案の集合、情報の集合、

A Knowledge Acquisition Agent for Decision Making
Using Information on the Internet.

Kazunori FUJIMOTO, (有) フジモト・リサーチパーク,
Fujimoto Research Park Co., LTD.

Yutaka YAMAMOTO, 京都大学大学院 情報学研究科,
Graduate School of Informatics, Kyoto University.

確率分布に着目する。そして、これらを獲得するそれぞれの要素を知識獲得エージェントとみなし、代替案獲得、情報獲得、確率分布獲得のエージェントについて、具体例をもとに説明する。

2.1 代替案獲得エージェント

代替案の集合は、意思決定の対象となる要素の集合であり、例えば、電化製品を購入する場合では、購入対象になる機種が代替案となる。代替案獲得エージェントは、ユーザから意思決定の対象のカテゴリ (例えば「デジタルカメラ」など) が与えられたとき、(1) カテゴリ名をキーワードとして代替案を集めたサイトを検索し、(2) サイトにアクセスしてページを収集し、(3) 集められたページから代替案の集合を抽出する、という過程で獲得作業を実施する。代替案を集めたサイトとしては、例えば、電化製品の選定では価格調査サイト (e.g., <http://www.kakaku.com/>)、政治家の選定では、議員要覧サイト (e.g., <http://www.seiji-koho.co.jp/>) などがある。

2.2 情報獲得エージェント

「情報」の集合は、ある代替案を選択した後の「状況」を予測するためのものである。デジタルカメラの購入の例では、この「情報」は、CCD 画素数などの仕様となる。こうした仕様は、機種の画質の良し悪しという「状況」を予測するのに利用できる。情報獲得エージェントは、代替案獲得エージェントの獲得した代替案について、(1) 代替案の名称をキーワードとして「情報」を集めたサイトを検索し、(2) サイトから「情報」に関するページを収集し、(3) 集められたページから「情報」の集合を抽出する、という過程で獲得作業を実施する。「情報」を集めたサイトとしては、例えば、電化製品の選定ではメーカーの提供する製品解説ページ、政治家の選定では各政治家のホームページなどがある。

2.3 確率分布獲得エージェント

確率分布は、「情報」の集合を使って、代替案を選択した後の「状況」を確率的に予測するものである。例えば、「情報」、「状況」として、それぞれ CCD 画

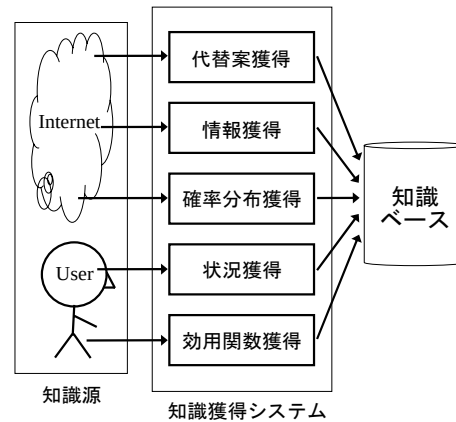


図 1 意思決定系での知識獲得システム

素数、画質の良し悪しが与えられたとき、この確率分布は、CCD 画素数から画質の良し悪しを予測する確率である。ネット上に直接確率の値が数値として記述されていることはない。そこで、確率分布獲得エージェントは、確率分布に関する知識 (評価知識と呼ぶ) を収集/獲得し、それらを統合して確率分布を導出する (この導出については [2] を参照されたい)。確率分布の獲得は、情報獲得エージェントの獲得した「情報」と、ユーザから与えられた「状況」について、(1) 情報と状況の名称をキーワードとして評価知識を提供するサイトを検索し、(2) サイトから評価知識に関するページを収集し、(3) 集められたページから評価知識を抽出し、(4) 獲得された評価知識を統合して確率分布を構成する、という過程で獲得作業を実施する。評価知識を提供するサイトとしては、例えば、電化製品の選定では、機種の評価ランキングを提供するサイト、あるいは、各機種の個人的な評価を提供する個人ページなどがある。

3 協調による知識獲得

本章では、2 章に述べた 3 つの知識獲得エージェントについて、容量の限られた通信路を共有しながら、知識の収集/獲得を行う協調獲得モデルを提案する。

3.1 エージェントの獲得戦略

本節では、知識獲得エージェントの獲得戦略として、知識ベースの価値を増す効果の大きい知識から

優先して収集する戦略を示す。こうした獲得戦略に基づくことにより、知識獲得エージェントは、価値の高い知識ベースをより早く構成することが可能となる。なお、この獲得戦略は各エージェントについて共通となることから、本節では、代替案、情報、確率分布を知識として区別せず、各知識の集合を同一の記号として E 、すでに獲得された知識ベースを KB、知識ベースの価値^{†1}を $V(KB)$ 、アクセスの対象となるアドレスの集合を R とそれぞれ表す。また、アドレスへのアクセスには、サイトを検索する行為と、ページを収集する行為が含まれるとする。

以上の設定のもと、エージェントがアドレス $r \in R$ へアクセスしたとき得られる知識ベースの価値の増加量の期待値は、得られる知識 $e \in E$ の見込みを表す確率 $P(e|r)$ を用いて次式で与えられる。

$$\sum_{e \in E} [V(KB \cup e) - V(KB)] \cdot P(e|r) \quad (1)$$

この期待値をアドレス r の**アクセス価値**と呼ぶ。アクセス価値の大きなアドレスほど、より大きな価値の知識ベースを期待できるという意味で、アクセスする価値があるといえる。こうした考えから、各エージェントは、知識の収集にあたって、得られる価値の期待値のより大きなアドレスを優先してアクセスする。

3.2 協調獲得モデル

協調獲得モデルの構成を図 2 に示す。協調獲得モデルは、知識源、通信路、3つの知識獲得エージェント、知識ベース、そして、アクセス価値ボードから構成される。アクセス価値ボードは、各エージェントが各知識のアクセス価値を共有するためのメモリである。こうした協調獲得モデルにおいて、各エージェントは、次の2つを判断しながら獲得作業を実施する。

- 各エージェント内で、どのアドレスへのアクセスが優先されるか。
- 他エージェントとの間で、どのエージェントが優先されるか。

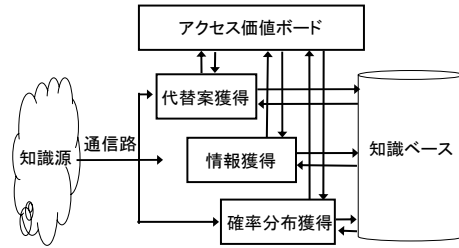


図 2 協調獲得モデル

以下では、通信路の容量制限から、ある時刻に通信路を利用できるのは1つのエージェントのみであるとして、効率よく知識獲得作業を実施するエージェントの振る舞いを示す。

1. すでに獲得されている知識ベースを調べ、対象となるアドレスへのアクセス価値を計算
2. アドレス価値の最大値を求め、その最大値を自エージェントの名称と合わせてアクセス価値ボードに提示
3. アクセス価値ボードを調べ、最大の価値を提示するエージェント名をサーチ
4. 最大の価値を提示するのが自エージェントである場合、(1) 収集を実行し、(2) 収集が完了したときアクセス価値ボードから自エージェントのアクセス価値を削除し、(3) 収集された情報から知識獲得を実行し、(4) 知識が獲得された場合は、知識ベースを更新して1.へ、獲得されない場合は、そのまま1.へ

こうした振る舞いにより、知識源への通信路の容量に制限があっても、高いアクセスの価値をもつ知識獲得エージェントが優先されながら、効率よく知識ベースを構成することが可能となる。

3.3 関連研究

統計的決定論のもとでの知識の価値については、代替案、確率分布が獲得されていることを前提に、「情報」の価値を議論することが多い[4][7]。これに対して、図1に示す意思決定系では、代替案、情報、確率分布を並行して獲得するので、「代替案の集合、情報の集合、確率分布」の3つの要素について、それぞれの価値を考える必要がある。例えば、代替案が全く獲

^{†1} 知識ベースの価値については、様々な定量化が考えられるが、その一例を付録に記す。

得されていない場合には、いくら情報や確率分布が獲得されても、利益 (効用) を得ることは期待できない。したがって、知識が全くない場合には、まず、代替案の獲得の価値が高いことになる。一方、代替案のみが獲得され、情報と確率分布が全く獲得されない場合には、適切な選択ができないので、大きな利益を得ることは期待できないことになる。したがって、代替案を獲得した後は、それを選ぶための知識の価値も高まることになる。このように、意思決定系では、「代替案の集合、情報の集合、確率分布」を区別せずに、それらの価値を議論することが重要となる。

情報の価値に基づく獲得戦略については、Value-Driven Information Gathering [3] でもその基本的な考え方が示されている。これに対し、我々の知識獲得モデルは、代替案、情報、確率分布の知識獲得エージェントの協調により知識獲得を実現するという点が大きく異なる。ネット情報から知識を獲得するにあたっては、収集した Web ページ中に知識が存在するかどうかを判定する技術や、表やテキストから知識を抽出する技術など、様々な各知識固有の獲得技術が必要となる。こうした各知識固有の獲得技術をエージェントとしてカプセル化することにより、他知識の獲得技術との独立性を確保でき、技術の追加/変更を容易に行うことが可能となる。また、各知識の獲得手続きの実行に、異なる計算機資源を用いることも容易となる。このように、我々は、提案の協調獲得モデルには多くの利点があると考えます。

4 おわりに

本稿では、インターネット上の情報から意思決定に必要な知識を獲得する機構について、これを複数の知識獲得エージェントの協調により実現する協調獲得モデルを提案した。今後は、アクセス価値の定義で用いた「アクセスの結果得られる知識の見込み」の設定法をはじめ、モデルの詳細化を進める。また、今回は、ある時刻に通信路を利用できるのは 1 つのエージェントのみであるとしたが、今後は、容量の限界まで複数のエージェントが利用できるアルゴリズムへの拡張を検討する。

参考文献

- [1] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from world wide web. In *AAAI-98*, pp. 509–516, 1998.
- [2] Kazunori Fujimoto, Kazumitsu Matsuzawa, and Hideto Kazawa. An elicitation principle of subjective probabilities from statements on the internet. In *Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems (KES-99)*, pp. 459–463, 1999.
- [3] Joshua Grass and Shlomo Zilberstein. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems*, Vol. 14, pp. 5–27, 2000.
- [4] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [5] Arnaud Sahuguet and Fabien Azavant. Wysiwyg web wrapper factory (w4f). Available from <http://db.cis.upenn.edu/W4F/>, 1999.
- [6] 市村, 長谷川, 渡部, 佐藤. テキストマイニング: 事例紹介. 人工知能学会誌, Vol. 16, No. 2, pp. 192–200, 2001.
- [7] 植野真臣. 意思決定アプローチによる bayesian network の因果モデル構築. 人工知能学会誌, Vol. 11, No. 5, pp. 725–734, 1996.
- [8] 藤本, 賀沢, 佐藤, 島津, 北. ネット情報を使った意思決定支援 dsu における知識獲得技術. 人工知能学会論文誌, Vol. 16, No. 1, pp. 120–129, 2001.
- [9] 藤本, 山本. ネット情報を使って意思決定する過程の数理モデルの提案. 人工知能学会研究会資料 SIG-FAI/KBS-J, pp. 29–33, 2001.
- [10] 富浦, 渡辺他. 特集: ここまできた自然言語処理. 情報処理学会誌, Vol. 41, No. 7, pp. 762–796, 2000.

付録: 知識ベースの価値

知識ベースの価値について、「知識ベースを利用した場合に期待できる効用」に基づいた定量化を示す。

定義 1 (知識ベースの価値)

知識ベースとして、代替案の集合 A 、情報の集合 E 、確率分布 Pr が獲得されたとき、

$$V(\{A, E, \text{Pr}\}) = \max_{a \in A} \sum_{\theta \in \Theta} U(\theta) \text{Pr}(\theta | (E : a)) \quad (2)$$

を知識ベース $\{A, E, \text{Pr}\}$ の価値と呼ぶ。ここに、 U は効用関数、 Θ は「状況」の集合、 $(E : a)$ は代替案 a についての情報の集合をそれぞれ表す。 □

情報財の開発インセンティブ

藤山英樹

This article explores competitive interaction in the development of new information goods. Fudenberg and Tirole (1986) model is extended to apply to the case of information goods duopoly market in which profits and prices depend on consumers' willingness to pay. The incentive to develop new information goods is assessed through calculated examples.

1 はじめに

情報財市場における巨大企業の代表である Microsoft は、独占禁止法違反をめぐり、司法省と争っている。^{†1}その独占力が反競争的な影響力を与えているかどうかが主要な論点である。さらに、Linux のように、オープンなソフトが大きな影響力を持ち始めている。しかし、Linux は未だ初心者にとって必ずしも扱いやすいものとは言い難く、Microsoft は次々とアプリケーションや OS をバージョンアップし、過剰ともいえるほどの高機能な情報財を開発し続けている。

競争の阻害が社会的な余剰を低下させることを論拠として、独占禁止法は制定されている。しかしながら、情報財は開発コストが大きい一方で、複製が容易であるという通常の財とはことなる特徴を持ち、こ

のような財の特質は、市場の発展において、財に独占力を一定程度認めることが重要となる。というのも、全くの同質財の競争においては、際限のない価格競争がもたらされるからである。

以上のように、通常の財と異なる性質をもつ情報財をめぐる競争において、企業間の競争と社会的な余剰の関係は必ずしも明らかではない。本稿では、Microsoft のような大企業とその他の多くの小企業における、新しい情報財の開発インセンティブを探ることとする。

本稿における情報財の市場とは、消費者の支払意思額にしたがい、一定の独占力のもと価格が決定される市場であり、情報財の開発は、既存の情報財に対する新機能の付加として扱われる。

複数企業間の競争における新しい情報財の開発は次の 2 つの効果をふまえた上で決定される。ひとつは、その情報財をより魅力的にすることによる消費者の需要を拡大もしくは市場シェアを拡大し、より大きな利潤を得る効果であり、もうひとつは、ライバル企業に先に情報財を開発されて市場のシェアを奪われことを防ぐという効果である (Tirole(1988))。独占的な立場にいる企業では、既に大きな利潤を挙げており、後者の効果が大きいであろうし、市場に新規参入しようという企業は後者の効果は存在せず前者の効果だけが影響を与えるであろう。

以上の 2 つの効果をふまえて、Fudenberg and Tirole (1986) ではコストを削減する新技術に関する開

An Incentive to Develop New Information Goods.
Hideki FUJIYAMA, 獨協大学経済学部, Dokkyo University, Faculty of Economics.

†1 一般に情報財はデジタル化され売買されるもの全てと定義される (Shapiro and Varian(1999))。

発競争モデルが提示された。^{†2}しかし、情報財の新開発ではコストの低下が利潤を拡大させるのではなく、消費者の支払意思額の増大が利潤を拡大させる。さらに、Fudenberg and Tirole (1986) では、コスト削減が利潤へ影響を与えるため、特に利潤最大化行動を明示的に示されることなく、利潤の変化を与えることができた。情報財の新開発では、支払意思額の増加が問題となり、利潤の変化を求めるためには利潤最大化行動を明示的に扱わなければならない。

本稿の目的は、新開発が利潤にあたえる変化を、支払意思額の変化と各企業の価格競争から導き、Fudenberg and Tirole (1986) モデルを拡張することである。さらに、そのモデルを用いて数値例を解き、大企業および小企業の情報財の開発インセンティブについての考察を深めることである。特に開発時期と社会的余剰の変化に注目する。

構成は以下の通りである。次節でモデルを提示し、第 3 節で数値例を用いてモデルを解く。第 3.1 節では大企業しか存在しない独占のケースを他のケースの比較基準として取り上げている。第 3.2 節では大企業が既存企業として存在し、小企業が市場への参入をうかがうケースを、第 3.3 節では大企業と小企業が共に市場に存在している複占のケースを考察している。第 4 節でまとめをおこなう。

2 モデル

2.1 利潤の決定

大企業と小企業の複占モデルを考える。2つの企業は同様の目的に用いられるが異なる情報財を生産している。大企業は多機能な情報財を、小企業は機能を限定した情報財を生産する。例えば、多機能なワープロとコンパクトなエディターとの関係である。消費者は $[0, 1]$ の区間に一様に分布している。支払意思額は線型に分布しており、相対的に 0 に近いほど大企業の情報財を、1 に近いほど小企業の情報財を選好するものとする。^{†3}それぞれの企業の財に対する支払意思額が図 1 によって示されている。大企業の情報財の方

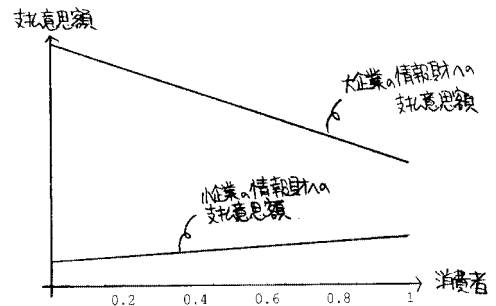


図 1 支払意思額の分布

が、より支払意思額が高くなると仮定している。^{†4}

この支払意思額を前提として各企業は価格付けをする。^{†5}大企業は市場のシェアを犠牲にしても、高価格を付けることによって、より高い利潤を得ることがある。このときに、全ての消費者において、支払意思額で劣っている小企業が市場でシェアを得ることが可能となる。

2.2 開発時期の決定

情報財の市場より、毎期ごとに利潤がフローとして得られるとする。新しい機能を既存の情報財に付加することによって、消費者の支払意思額を高めることが出来る。

しかし、より早い開発はより多くの開発コストがかかるものとする。つまり、技術開発水準の向上といった学習効果、他の目的で開発された技術を流用することといった外部効果が働いており、時間の経過と共に開発コストが小さくなると仮定する。ここでは、単純に、 c をある定数、 t を時間として、

$$\frac{c}{t} \quad (1)$$

^{†4} このような支払意思額の分布が妥当するのは、OS やワープロソフトなどのソフトウェアの競合があげられる。そこで生じがちなネットワーク外部性も大企業の情報財への支払意思額の大きさに示されることになる。

^{†5} 情報財におけるこのような価格付けの重要性は Shapiro and Varian(1999) をはじめ広く指摘されている。

^{†2} 第 3 章 (b) を見よ。

^{†3} このような線形の支払意思額の分布の仮定は Chung-I Chuang and Sirbu(2000) でも用いられている。

と定式化する。

また、既存企業の開発コストと参入企業の開発コストも同一とする。^{†6}

お互いの企業が、コスト関数が既知もと、利子率を $r\%$ として、利潤の流列の割引現在価値が最大となるように、新機能の開発時期に関する計画を立てる。^{†7}

以上により、Fudenberg and Tirole (1986) の新技術開発競争モデルが、支払意思額に基づいた情報財の新機能開発競争モデルに拡張された。

3 数値例による分析

数値例でモデルを解き、情報財の開発競争が過剰となり、開発競争が社会的厚生を低める場合があることを示す。

3.1 大企業の独占：基準ケース

消費者は区間 $[0, 1]$ に一様分布している。ある消費者 ($x \in [0, 1]$) における大企業の情報財への消費者の支払意思額 ($V_1(x)$) を

$$V_1(x) = 20 - 10x \quad (2)$$

とする。このとき、大企業は価格を 10 とすることで、利潤 (π_{0m}) を最大にすることができ、利潤は 10 となる。

新機能の開発による支払意思額の増加については、ある消費者 ($x \in [0, 1]$) における大企業の情報財への消費者の支払意思額 ($V_1(x)$) を

$$V_1(x) = 22 - 12x \quad (3)$$

と仮定する。^{†8} このとき、大企業は価格を 11 とすることで、利潤 (π_{1m}) を最大にすることができ、利潤は $\frac{121}{12} (\simeq 10.08)$ となる。

コスト関数については、期初において開発のコストが十分に大きくなるように

$$c = 12 \quad (4)$$

とする。さらに、利子率 (r) は 0.05 とし^{†9}、開発時期を i とする。

以上より、利潤の流列の現在価値は

$$\pi = \int_0^i \pi_{0m} \cdot e^{-rt} dt + \int_i^\infty \pi_{1m} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} \quad (5)$$

となり、最大化の 1 次条件である

$$\frac{\partial \pi}{\partial i} = 0 \quad (6)$$

を解くと、 $i \simeq 16.12$ となる。^{†10} このときの利潤、つまり生産者余剰、は $\pi \simeq 200.41$ である。消費者余剰は 100.37 となり、社会的総余剰は 300.78 となる。

社会的に最適となるのは、つまり、生産者余剰および消費者余剰を合わせて最大化するとき、 $i \simeq 5.07$ となり、このとき、生産者余剰は 199.46、消費者余剰は 107.75、社会的総余剰は 307.21 となる。

3.2 既存企業が大企業の小企業参入モデル

大企業を既存企業、小企業を参入企業とする、参入モデルを考察する。ここでは、各企業共に相手企業の反応を考慮した上で最適な開発時期を決定しなければならない。

大企業の情報財に対する支払意思額は前節と同様にある消費者 ($x \in [0, 1]$) に対して、式 (2) と仮定する。大企業が新機能を開発したときも前節と同様にある消費者 ($x \in [0, 1]$) に対して、式 (3) と仮定する。

小企業が参入していない時に得ていた大企業の利潤は、前節と同様に、 $\pi_{0m} = 10$ となる。小企業が参入していない時に新機能を開発したときの大企業の利潤も、前節と同様に、 $\pi_{1m} = \frac{121}{12} (\simeq 10.08)$ となる。

次に参入企業が新機能を開発し、情報財市場に参入してきたケースを考える。参入企業が新機能を開発し情報市場に参入した場合のある消費者 ($x \in [0, 1]$) における小企業の情報財への支払意思額 ($V_2(x)$) は

^{†6} かなり強い仮定であるが、もし、他の企業が開発した技術を買収する場合は、この仮定は満たされる。

^{†7} もしくは新機能の技術を買収する最適な時期を決定する。

^{†8} つまり、より評価していた消費者にはより高い支払意思額の増加がなされる。

^{†9} 以後全て c と r についてはこの値を与えている。

^{†10} 関数の形状により最大化のための 2 次条件も満たされている。

$$V_1(x) = 2x \quad (7)$$

であると仮定する。

参入した時点では既存企業が新機能をまだ開発していないケースの利潤を求める。つまり、既存企業である大企業と参入企業である小企業のそれぞれの情報財への支払意思額が式 (2)、式 (7) となっているときの均衡での利潤を求める。^{†11} 結果は次の通りである：大企業の価格が $\frac{32}{3} (\simeq 10.66)$ および小企業の価格が $\frac{4}{3} (\simeq 1.33)$ であり、大企業の利潤 (π_{1d}) が $\frac{256}{27} (\simeq 9.48)$ および小企業の利潤 (π_{2d}) が $\frac{4}{27} (\simeq 0.14)$ である。消費者余剰は $\frac{392}{81} (\simeq 4.83)$ となり、社会的総余剰は 14.45 となる。大企業のシェアは $\frac{8}{9} (\simeq 0.88)$ である。

次に、参入した時点で既存企業が新機能を開発しているケースの利潤を求める。つまり、既存企業である大企業と参入企業である小企業のそれぞれの情報財への支払意思額が式 (3)、式 (7) となっているときの均衡は次の通りである。大企業の価格が $\frac{607}{52} (\simeq 11.67)$ および小企業の価格が $\frac{179}{104} (\simeq 1.721)$ であり、大企業の利潤 (π_{1dw}) が $\frac{108653}{10816} (\simeq 10.04)$ および小企業の利潤 (π_{2dw}) が $\frac{5191}{21632} (\simeq 0.23)$ である。消費者余剰は $\frac{193087}{43264} (\simeq 4.46)$ となり、社会的総余剰は 14.73 となる。大企業のシェアは $\frac{179}{208} (\simeq 0.86)$ である。^{†12}

Fudenberg and Tirole (1986) にしたがって、

1. 既存企業について Leader-Follower 曲線を描く。
2. 参入企業について Leader-Follower 曲線を描く。
3. 二つの Leader-Follower 曲線を比較する。

という手順にしたがい Nash 均衡を求める。

3.2.1 既存企業の Leader-Follower 曲線

はじめに既存企業の Leader 曲線を得る。つまり、 i 期に Leader として新技術を開発し、その後、参入

企業が n 期に技術を開発したときの既存企業の利潤関数を求める。ここで、既存企業が Leader として新技術を開発した時期 (i) は外生的に与えられるが、その後の参入企業の開発時期 (n) はそれを所与として最適なものが内生的に得られる。

ここでは利潤の流列の割引現在価値の和が問題となるが、この利潤の流列の和についての利潤関数を、既存企業については π_1 と、参入企業については π_2 と表記する。それぞれ、

$$\begin{aligned} \pi_1 = & \int_0^i \pi_{0m} \cdot e^{-rt} dt + \int_i^n \pi_{1m} \cdot e^{-rt} dt \\ & + \int_n^\infty \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} \end{aligned} \quad (8)$$

$$\pi_2 = \int_n^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} \quad (9)$$

となる。

以下、後ろ向き帰納法で解いてゆく。後から技術を開発する参入企業の最適な開発時期は、式 (9) を解いて、 $n \simeq 8.43$ が得られるが、ここでは、前提として既存企業が Leader、つまり、 $i \leq n$ となるので、

$$n = \max\{8.43, i\} \quad (10)$$

が得られる。つまり、8.43 より以前に既存企業が Leader として新技術を開発したならば、参入企業は 8.43 まで待って、新技術を開発し、他方、8.43 より以降に既存企業が Leader として新技術を開発したならば、参入企業はすぐに、つまり同時に、新技術を開発することになる。^{†13}

参入企業の以上のような行動を前提として、 i 期に既存企業が Leader として新技術を開発するときの利潤関数は、

^{†11} 各企業の最適反応関数およびその交点から均衡は得られる。

^{†12} 以上のような数値例は、既存企業である大企業が高価格で高利潤を、参入企業である小企業が低価格で低利潤を得るという関係をしめしており、現実には、ひとつの企業が巨大なシェアを持つ市場に小さな企業が参入した後に生じるとされる大小関係とひどくかけ離れてはいないであろう。

^{†13} すぐに技術を開発することは利潤関数の形状からわかる。実際には意思決定がなされるのは期初だけであるが、そのような思考実験がなされるということである。技術を開発した企業を買収するケースでは期ごとに意思決定が行われると解釈することもできる。

$$\pi_1 = \begin{cases} \int_0^i \pi_{0m} \cdot e^{-rt} dt + \int_i^{8.43} \pi_{1m} \cdot e^{-rt} dt \\ + \int_{8.43}^{\infty} \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} & \text{if } i \leq 8.43 \\ \int_0^i \pi_{0m} \cdot e^{-rt} dt + \int_i^{\infty} \pi_{1dw} \cdot e^{-rt} dt \\ - \frac{c}{i} \cdot e^{-ri} & \text{if } i > 8.43 \end{cases} \quad (11)$$

となる。これより、既存企業の Leader 曲線が描ける。^{†14}

次に Follower 曲線を得る。つまり、参入企業が n 期に新技術を開発したときの、Follower としての利潤関数を求める。^{†15}

ここにおいて、この利潤の流列の和に関しての既存企業の利潤関数 (π_1)、参入企業の利潤関数 (π_2) はそれぞれ、

$$\pi_1 = \int_0^n \pi_{0m} \cdot e^{-rt} dt + \int_n^i \pi_{1d} \cdot e^{-rt} dt + \int_i^{\infty} \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} \quad (12)$$

$$\pi_2 = \int_n^i \pi_{2d} \cdot e^{-rt} dt + \int_i^{\infty} \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} \quad (13)$$

となる。

以下、後ろ向き帰納法で解いてゆく。後から技術を開発する既存企業の最適な開発時期は、式 (12) を解いて、 $i \simeq 5.17$ が得られるが、ここでは、前提として参入企業が Leader、つまり、 $n \leq i$ となるので、

$$i = \max\{5.17, n\} \quad (14)$$

が得られる。つまり、5.17 より以前に参入企業が Leader として新技術を開発したならば、既存企業は 5.17 まで待って、新技術を開発し、他方、5.17 より以降に参入企業が Leader として新技術を開発したな

^{†14} π はすべて前節に求めた値を用いている。

^{†15} ここでも、参入企業が Leader として新技術を開発した時期 (n) が外生的に与えられる、その後の既存企業の開発時期 (i) を、それ (n) を所与として内生的に得る。

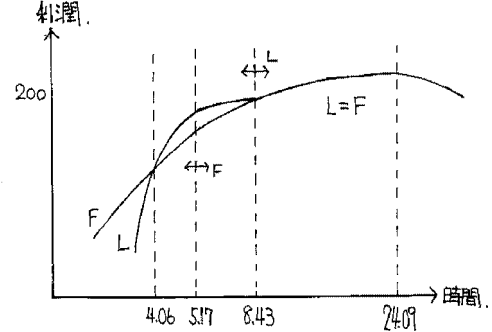


図 2 既存企業 Leader-Follower 曲線 (既存企業が大企業である。)

らば、既存企業はすぐに、つまり同時に、新技術を開発することになる。

以上より、 n 期に参入企業が Leader として新技術を開発するときの Follower としての既存企業の利潤関数は、

$$\pi_1 = \begin{cases} \int_0^n \pi_{0m} \cdot e^{-rt} dt + \int_n^{5.17} \pi_{1d} \cdot e^{-rt} dt \\ + \int_{5.17}^{\infty} \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{5.17} \cdot e^{-r \cdot 5.17} & \text{if } n \leq 5.17 \\ \int_0^n \pi_{0m} \cdot e^{-rt} dt + \int_n^{\infty} \pi_{1dw} \cdot e^{-rt} dt \\ - \frac{c}{n} \cdot e^{-rn} & \text{if } n > 5.17 \end{cases} \quad (15)$$

となる。これより、既存企業の Follower 曲線が描ける。

以上をまとめて、既存企業の Leader-Follower 曲線は図 2 にまとめられている。初期においては、Follower となるほうがより高い利潤を得るが、4.06 以降は Leader となる方がより高い利潤を得ることとなり、8.43 以降で Leader 曲線と Follower 曲線は一致することとなる。

3.2.2 参入企業の Leader-Follower 曲線

前節と全く同様に参入企業の Leader 曲線を求める。 n 期に Leader として新技術を開発し、その後、既存企業が i 期に技術を開発したときの利潤関数を求める。^{†16}

^{†16} ここで、参入企業が Leader として新技術を開発した時期 (n) は外生的に与えられるが、その後の既存企業の開発時期 (i) はそれを所与として最適なもの

ここでも利潤の流列の割引現在価値の和が問題となるが、この利潤の流列の和に関しての利潤関数を改めて、既存企業については π_1 と、参入企業については π_2 と表記する。それぞれ、

$$\pi_2 = \int_n^i \pi_{2d} \cdot e^{-rt} dt + \int_i^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} \quad (16)$$

$$\pi_1 = \int_0^n \pi_{0m} \cdot e^{-rt} dt + \int_n^i \pi_{1d} \cdot e^{-rt} dt + \int_i^\infty \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} \quad (17)$$

となる。

以下、後ろ向き帰納法で解いてゆく。後から技術を開発する既存企業の最適な開発時期は、式 (17) を解いて、 $i \simeq 5.17$ が得られるが、ここでは、前提として参入企業が Leader、つまり、 $n \leq i$ となるので、

$$i = \max\{5.17, n\} \quad (18)$$

が得られる。つまり、5.17 より以前に参入企業が Leader として新技術を開発したならば、既存企業は 5.17 まで待って、新技術を開発し、他方、5.17 より以降に参入企業が Leader として新技術を開発したならば、既存企業はすぐに、つまり同時に、新技術を開発することになる。

既存企業の以上のような行動を前提として、 n 期に参入企業が Leader として新技術を開発するときの利潤関数は、

$$\pi_2 = \begin{cases} \int_n^{5.17} \pi_{2d} \cdot e^{-rt} dt + \int_{5.17}^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} & \text{if } n \leq 5.17 \\ \int_n^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} & \text{if } n > 5.17 \end{cases} \quad (19)$$

となる。これより、参入企業の Leader 曲線が描ける。

次に Follower 曲線を得る。つまり、既存企業が i 期に新技術を開発したときの、Follower として得られる利潤を求める。^{†17}

^{†17} が内生的に得られる。

^{†17} ここで、既存企業が Leader として新技術を開発し

ここにおいて、この利潤の流列の和に関しての既存企業の利潤関数 (π_1)、参入企業の利潤関数 (π_2) はそれぞれ、

$$\pi_2 = \int_n^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{n} \cdot e^{-rn} \quad (20)$$

$$\pi_1 = \int_0^i \pi_{0m} \cdot e^{-rt} dt + \int_i^n \pi_{1m} \cdot e^{-rt} dt + \int_n^\infty \pi_{1dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} \quad (21)$$

となる。

以下、後ろ向き帰納法で解いてゆく。後から技術を開発する参入企業の最適な開発時期は、式 (20) を解いて、 $n = 8.43$ が得られるが、ここでは、前提として参入企業が Leader、つまり、 $i \leq n$ となるので、

$$n = \max\{8.43, i\} \quad (22)$$

が得られる。つまり、8.43 より以前に既存企業が Leader として新技術を開発したならば、参入企業は 8.43 まで待って、新技術を開発し、他方、8.43 より以降に参入企業が Leader として新技術を開発したならば、既存企業はすぐに、つまり同時に、新技術を開発することになる。

以上より、 i 期に既存企業が Leader として新技術を開発するときの Follower としての参入企業の利潤関数は、

$$\pi_2 = \begin{cases} \int_{8.43}^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{8.43} \cdot e^{-r8.43} & \text{if } i \leq 8.43 \\ \int_i^\infty \pi_{2dw} \cdot e^{-rt} dt - \frac{c}{i} \cdot e^{-ri} & \text{if } i > 8.43 \end{cases} \quad (23)$$

となる。これより、参入企業の Follower 曲線が描ける。

以上をまとめて、参入企業の Leader-Follower 曲線は図 3 にまとめられている。初期においては、Follower となるほうがより高い利潤を得ることができ、8.43 期以降は、Leader 曲線と Follower 曲線は一致す

た時期 (i) が外生的に与えられる、その後の参入企業の開発時期 (n) を、それ (i) を所与として内生的に得る。

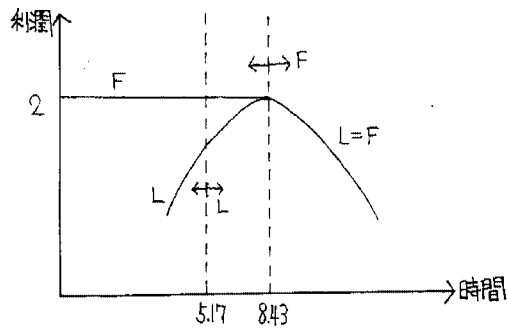


図 3 参入企業 Leader-Follower 曲線 (既存企業が大型企业である)

る。全体として, Follower となることが Leader となることを弱支配している。

3.2.3 Nash 均衡

均衡においては, 両企業が 8.43 期に同時に新開発する。両企業とも, なるべく開発を遅くして利潤を高めようとする。しかし, 参入企業は 8.43 期以降利潤は低下してしまうので, 8.43 期に開発を決定する。それを受けて, 既存企業も 8.43 期に開発することとなる。このとき, 既存企業の利潤 (余剰) は 199.66, 参入企業の利潤 (余剰) は 2.21 となり, 生産者余剰は 201.87 となる。消費者余剰は 92.95 であり, 社会的総余剰は 294.83 となる。

3.1 節の独占のケースとの比較によって参入の効果を検討すると, 既存企業の利潤は低下するが, 生産者全体としての利潤 (余剰) は高まる。他方, 消費者余剰は低下し, 社会的総余剰も低下する。つまり, 社会全体としては独占状態の方が望ましいことがこの例では示されている。

3.3 大企業と小企業の複占モデル

ここでは市場に既に大企業と小企業が存在する複占モデルを考察し, 独占モデルとの比較を行う。

大企業の情報財に対する支払意思額は前節までと同様にある消費者 ($x \in [0, 1]$) に対して, 式 (2) を仮定する。大企業が新機能を開発したときも前節までと同様にある消費者 ($x \in [0, 1]$) に対して, 式 (3) を仮定する。

既に市場に存在する小企業の情報財に対する支払意思額は前節の参入後の小企業の場合と同様にある消費者 ($x \in [0, 1]$) に対して, 式 (7) を仮定する。小企業が新機能を開発したときはある消費者 ($x \in [0, 1]$) に対して,

$$V_1(x) = 4x \quad (24)$$

を仮定する。

両企業とも新機能を開発する前の, つまり, 大企業と小企業のそれぞれの情報財への支払意思額が式 (2), 式 (7) となっているときの均衡は次の通りである。大企業の価格が $\frac{32}{3} (\approx 10.66)$ および小企業の価格が $\frac{4}{3} (\approx 1.33)$ であり, 大企業の利潤が $\frac{256}{27} (\approx 9.48)$ および小企業の利潤が $\frac{4}{27} (\approx 0.14)$ である。消費者余剰は $\frac{392}{81} (\approx 4.83)$ となり, 社会的総余剰は 14.45 となる。大企業のシェアは $\frac{8}{9} (\approx 0.88)$ である。

大企業だけが開発したときの, つまり, 大企業と小企業のそれぞれの情報財への支払意思額が式 (3), 式 (7) となっているときの均衡は次の通りである。大企業の価格が $\frac{607}{52} (\approx 11.67)$ および小企業の価格が $\frac{179}{1644} (\approx 1.72)$ であり, 大企業の利潤が $\frac{108653}{10816} (\approx 10.04)$ および小企業の利潤が $\frac{5159}{21632} (\approx 0.23)$ である。消費者余剰は $\frac{193087}{43264} (\approx 4.46)$ となり, 社会的総余剰は 14.73 となる。大企業のシェアは $\frac{179}{208} (= 0.86)$ である。

小企業だけが開発したときの, つまり, 大企業と小企業のそれぞれの情報財への支払意思額が式 (2), 式 (24) となっているときの均衡は次の通りである。大企業の価格が $\frac{34}{3} (\approx 11.33)$ および小企業の価格が $\frac{8}{3} (\approx 2.66)$ であり, 大企業の利潤が $\frac{578}{63} (\approx 9.17)$ および小企業の利潤が $\frac{32}{63} (\approx 0.50)$ である。消費者余剰は $\frac{247}{63} (\approx 3.92)$ となり, 社会的総余剰は 14.59 となる。大企業のシェアは $\frac{17}{21} (\approx 0.80)$ である。

両企業が開発したときの, つまり, 大企業と小企業のそれぞれの情報財への支払意思額が式 (3), 式 (24) となっているときの均衡は次の通りである。大企業の価格が $\frac{437}{35} (\approx 12.48)$ および小企業の価格が $\frac{11}{35} (\approx 3.17)$ であり, 大企業の利潤が $\frac{48507}{4900} (\approx 9.89)$ および小企業の利潤が $\frac{3219}{4900} (\approx 0.65)$ である。消費者余剰は $\frac{9451}{2450} (\approx 3.85)$ となり, 社会的総余剰は 14.39 となる。大企業のシェアは $\frac{111}{140} (\approx 0.79)$ である。

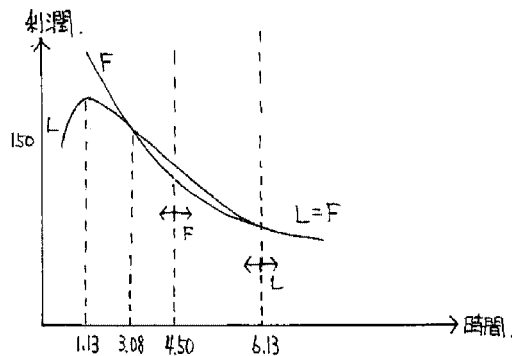


図 4 大企業 Leader-Follower 曲線

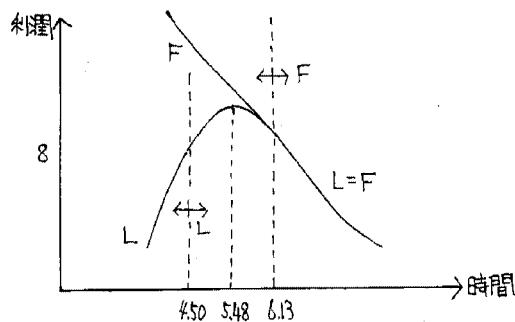


図 5 小企業 Leader-Follower 曲線

3.3.1 Leader-Follower 曲線

前節同様にそれぞれの企業の Leader-Follower 曲線を求めることが出来る。大企業の Leader-Follower 曲線は図 4 で示されている。小企業の Leader-Follower 曲線は図 5 で示されている。

3.3.2 Nash 均衡

均衡では、1.13 期に大企業が新開発し、6.13 期に小企業が新開発する。既存企業の利潤（余剰）は 177.85、参入企業の利潤（余剰）は 9.39 となり、生産者余剰は 187.24 となる。消費者余剰は 80.76 であり、社会的総余剰は 268.00 となる。

3.1 節の独占のケースとの比較によって複占状態の効果を考察すると、生産者余剰、消費者余剰、社会的総余剰の全てにおいて余剰は低下し、参入のケースと同様に、社会全体としては独占状態の方が望ましいこ

とがこの例では示されている。^{†18}

最後にこの節の各結果を表 1 にまとめておく。

4 結び

本稿では、Fudenberg and Tirole (1986) のモデルを拡張し、参入のケースおよび複占のケースに関しての情報財に対する開発インセンティブを数値例によって考察した。

参入のケースにおいては、大企業が独占のケースと比較して、開発時期は早まり、小企業と大企業が同時に開発する。複占のケースでは、大企業の独占のケースと比較して、さらに、開発時期は早められた。しかしながら、社会的総余剰は、独占のケース、参入のケース、複占のケースという順に、低下していった。

他の支払意思額の変化を仮定すると、もちろん他の結論も得られる。例えば、大企業の支払意思額の変化が全ての消費者に同様に起こる場合では、参入のケースでは、情報財の開発の時期は変化しないが、小企業は技術開発コストが十分に低下してから参入をするという例が得られており、複占のケースでは、前節の数値例と同様に大企業の技術開発が早まるケースも得られている。以上の場合でも、競争の存在が社会的余剰を低下させている。

参入のケースにおいて、需要の価格弾力性が大きいときには、既存大企業が参入を阻止するような均衡も得られている。この場合も、参入することによる厚生損失と、参入阻止による厚生損失の大小によって、独占企業の価格戦略を評価しなければならない。

このように、支払意思額に基づいた新しい情報財の開発競争においては、複数企業の開発競争が社会的厚生を増大させるについては、十分に注意をして判断しなければならない。

ただ、競争が開発競争を促すことから、技術革新を押し進めるという基準のもとでは、より競争を取り入れることが有効となろう。

今後の課題として、数値例ではなく各パラメータの関係から新しい情報財の開発インセンティブを比較

^{†18} ここでの複占のケースは消費者の支払意思額がより大きくなっているのに、社会的余剰は低下し、より効率性を損なっているといえる。

表 1 各ケースでの開発時期と余剰

	計画	独占	参入	複占
開発時期	5.07	16.12	8.43(両)	1.13+6.13 (大) (小)
生・余剰	199.46	200.41	201.87	187.24
消・余剰	107.75	100.37	92.95	80.76
総・余剰	307.21	300.78	294.82	268.00

することがある。また、支払意思額の分布に関して、および新しい情報財の開発がどのように支払意思額を高めるかに関して、より詳細な考察が求められる。

謝辞

様々な情報財に関する様々な研究機会をご紹介いただいている京都大学大学院情報学研究科石田亨教授、また、ワークショップでコメントをいただいた方々に深く感謝いたします。当研究は文部科学省科学研究費補助金を受けている。

参考文献

- [1] Chung-I Chuang, J., and M. A. Sirbu. 2000. Network Delivery of Information Goods: Optimal Pricing of Articles and Subscriptions. Internet publishing and beyond ed. Kahlin, B and Varian, H. R. The MIT Press, Cambridge, Massachusetts.
- [2] Fudenberg, D., and J. Tirole. 1986. Dynamic Models of Oligopoly. Volume 3 of Fundamentals of Pure and Applied Economics, ed. J. Lesourne and H. Sonnenschein. London: Harwood.
- [3] 藤山英樹, 八槇博史, 石田亨, 2001. 「情報財と情報市場戦略」『情報処理』Vol. 42 No. 4. pp.410-414. 情報処理学会.
- [4] Shapiro, C., and Varian, H.R. 1999. Information Rules., Harvard business school press, Boston, Massachusetts. (千本倅生訳:「ネットワーク経済」の法則, IDG ジャパン).
- [5] Tirole, J. 1988. The Theory of Industrial Organization., MIT Press. Cambridge.

人工市場における複雑さの発展の分析

和泉 潔

In this study we rethought efficient market hypothesis from a viewpoint of complexity of market participants' prediction methods and market price's dynamics, and examined the hypothesis using simulation results of our artificial market model. As a result, we found the two difference from the hypothesis. (a) Complexity of markets was not fixed, but changed with complexity of agents. (b) When agents increased the complexity of their prediction methods, structure of dynamic patterns of market price didn't disappear, but it can't be described by equation of any dimensions.

1 はじめに

専門家の行う金融価格の予測と、目隠した猿がダーツに矢を投げてした予測が同じであると言ったらビックリするであろうか。

経済学の分野では 70 年代に効率的市場仮説という金融市場の理論が提唱され、今日まで大きな議論的となっている。この仮説では金融市場の価格の挙動はランダムウォークであり予測不可能であるとしている。そのため、最初に言ったようなことが言われるのである。今日、大きく発展した金融工学の理論もこの仮説をベースにして、金融価格を確率過程で表すことから始まっている。

仮説の統計的な検証も実証データを用いて数多く行われたが、仮説が市場参加者の予想形成といったことを含

んでいるため、なかなか直接的に検証できなかった。しかし近年、計算機の中に仮想的な市場モデルを構築してシミュレーションを行う人工市場アプローチが現われ、これにより仮説の検証を直接的に行うという研究が出てきた [1-3]。

本研究は、市場参加者個人の予測方式の複雑さと市場価格の挙動の複雑さの関係という新しい視点から効率的市場仮説を捕らえ直し、人工市場モデルを用いたシミュレーション結果から、仮説の検討を行った。

2 複雑さから見た効率的市場仮説

効率的市場仮説の要点をまとめると下記のようなになる。

- 金融市場の市場参加者は市場価格の挙動に関係する全ての情報を、極めて迅速かつ的確に取り入れて、価格予想に用いる。
- そのような市場参加者間の取引で決定される市場価格は、現在までに利用可能で重要な情報を全て適切に反映している。
- 従って、ある者が現在までに利用可能な情報と市場価格の新たな関係を見つけ出して、他の者より有利になる余地はない。つまり市場価格の挙動はだれにも予測不可能な新規情報によって動くランダムウォークになる。

上記の要点を複雑さの観点から見ると、効率的市場仮説は下記のことを暗に含んでいる。

- 各市場参加者は適切な情報の取り込み方を得るために、自分の予測方式を学習により複雑にしていって、市場価格の決定式の構造をつかもうとする。

Kiyoshi IZUMI, 産業技術総合研究所, AIST & 科技団 さきがけ, PRESTO, JST, 電子メール: (kiyoshi@ni.aist.go.jp)

- 価格決定式の構造は市場参加者の学習とは独立して固定して存在し、最終的には市場参加者に構造を見破られて、その構造が消えてしまう。

つまり、各市場参加者の予測式の複雑さと市場価格の挙動の複雑さの独立が暗黙の前提となっているのである。

これに対し、de la Maza [4] は人工市場シミュレーションにより、市場参加者の予測式の次元が 0 から 1 に上がると、市場価格もランダムウォークから線形性を持つ挙動になることを見つけた。つまり、個人の複雑さと市場の複雑さが独立でない可能性を示した。

それでは、各市場参加者が自分の予測方程式をより複雑にする動機はどこにあるのだろうか。Joshi ら [6] によると、囚人のジレンマゲームに似た状況が起きているからだとしている。彼女らの人工市場モデルでは、予測方式にテクニカル分析の移動平均の手法を取り込んで、予測式の次元を 0 から 1 に上げることを、囚人のジレンマゲームの裏切り戦略に対応させた。これに対し、テクニカル分析を予測に用いないことを協調戦略に対応させた。シミュレーション結果より、囚人のジレンマ状況となる下記の 2 つの条件が見られた。

条件 1 予測次元を上げること (裏切り戦略) によって、予測の精度が上がって取引結果の収益が増大するメリットが得られた。

条件 2 しかし、皆が次元を上げると市場価格の挙動が複雑になり、皆がテクニカル分析を用いていなかった (協調戦略) とよりも予測精度が下がってしまった。このように、皆が利益追求のため自分の予測式の次元を上げようとして、結局は予測精度が下がってしまう状況が起きる。

本研究では次節以降で、人工市場シミュレーションにより、予測次元がより大きくなったら囚人のジレンマ的状況や市場の複雑さはどうなるのかについて分析を行う。その結果に基づき、今までの効率的市場仮説の検討を行う。

3 人工市場モデル

50 人の仮想ディーラー (エージェント) がいる人工市場を構築した。この人工市場には 1 つの金融商品と 1 つの無リスク資本が存在する。各エージェントは金融商品の価格の推移を予想して、予想収益の効用が最大になるように、金融商品と無リスク資本の持高を変える。人工市

場の中では、予想・注文・価格決定・学習のステップから 1 期間となり、この 4 つのステップを繰り返すことによって離散的に時間が進んでいく。

3.1 予想

今期の金融価格の変動値を、過去の金融価格の変動値の線形和で予想する。つまり、今回の研究では市場にファンダメンタルズ情報は存在せず、テクニカル分析のみによる予想となる。

各エージェントが持つ予想方程式は、自己回帰和分平均移動モデル ARIMA(n,1,0) である。n は何期前までの価格変動を予想に用いるかを表す。n が大きいほど予想方程式の次元が大きいことになるので、本研究ではこれを各エージェントの予想の複雑さで見なす。

P_t をまだ決定されていない今期の金融価格とし、 $\tilde{y}_t$ を金融価格の変動 ($P_t - P_{t-1}$) の予想値とすると、予想式は下記ようになる。

$$\begin{aligned}\tilde{y}_t &= \sum_{i=1}^n b_i y_{t-i} + e_t \\ &= \mathbf{x}_t' \mathbf{b}_t + e_t\end{aligned}\quad (1)$$

ただし、 e_t は平均 0、分散 0.1 の正規分布、 $\mathbf{b}_t$ は予測式の係数をまとめたベクトル $(b_1, \dots, b_n)'$ であり^{†1}、 $\mathbf{x}_t$ は予測式の説明変数、つまり過去の価格変動のベクトル $(y_{t-1}, \dots, y_{t-n})'$ である^{†2}。

3.2 注文

各エージェントはリスク回避型の予想収益の効用関数を持っているとする。このとき、効用が最大になるような金融商品の持高の最適値 q_t^* は、式 (1) の変動の予想値 y_t に比例する。

$$q_t^* = a y_t \quad (2)$$

今までに持っている金融商品持高を q_{t-1} とすると、各エージェントの注文量 o_t は最適持高との差になる。

$$o_t = q_t^* - q_{t-1} \quad (3)$$

†1 各エージェントの予測式の係数の初期値 $\mathbf{b}_0$ は -1 から 1 までの一様乱数によって与えられる。

†2 一番最初の期では $\mathbf{x}_0$ に平均 0、分散 1 の正規分布で生成された価格の時系列データを用い、その後の時期では $\mathbf{x}_t$ に人工市場で生成された価格データを用いた。

注文量が正(負)の場合は, エージェントが買い注文(売り注文)をすることを意味するので, 市場価格 P_t が自分の予想価格 ($P_{t-1} + y_t$) よりも安くて(高くて) 有利な場合は, o_t だけ買い注文(売り注文)をする.

$o_t > 0$ の場合

$$\begin{cases} o_t \text{ だけ買い} & (P_t \leq P_{t-1} + y_t \text{ のとき}) \\ \text{何もしない} & (P_t > P_{t-1} + y_t \text{ のとき}) \end{cases}$$

$o_t < 0$ の場合

$$\begin{cases} \text{何もしない} & (P_t < P_{t-1} + y_t \text{ のとき}) \\ o_t \text{ だけ売り} & (P_t \geq P_{t-1} + y_t \text{ のとき}) \end{cases}$$

3.3 価格決定

50 個のエージェントの売買戦略を市場全体で集積して, 需要と供給が均衡するような値にモデルの今期の市場価格が決定される. 市場価格よりも高い値段をつけた買い手と安い値段の売り手の間で取引が成立する.

3.4 学習

各エージェントは新たに決定された市場価格の変動 y_t の情報と逐次最小二乗法を用いて, 予測式 (1) の係数 $\mathbf{b}_t$ を更新する^{†3}. 係数の逐次更新の式は

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \frac{(\mathbf{X}_t' \mathbf{X}_t)^{-1} \mathbf{x}_t (y_t - \mathbf{x}_t' \mathbf{b}_t)}{f_t} \quad (4)$$

と表現される [5]. ただし, $\mathbf{X}_t$ は $\mathbf{X}_0 = 100 \times \mathbf{I}$ ($\mathbf{I}$ は単位行列) で始まる学習の行列であり, 下記の式で更新される.

$$(\mathbf{X}_t' \mathbf{X}_t)^{-1} = \frac{(\mathbf{X}_{t-1}' \mathbf{X}_{t-1})^{-1} - \frac{(\mathbf{X}_{t-1}' \mathbf{X}_{t-1})^{-1} \mathbf{x}_t \mathbf{x}_t' (\mathbf{X}_{t-1}' \mathbf{X}_{t-1})^{-1}}{f_t}}{f_t} \quad (5)$$

また,

$$f_t = 1 + \mathbf{x}_t' (\mathbf{X}_{t-1}' \mathbf{X}_{t-1})^{-1} \mathbf{x}_t \quad (6)$$

とする.

4 シミュレーション結果

人工市場モデルを用いて, 予測次元が大きくなったら囚人のジレンマの状況や市場の複雑さはどうなるのかシミュレーションで確かめた.

4.1 予測式を複雑にするメリット

予測式を複雑にした方が有利になるかどうか確かめた. 25 個のエージェントの予測式の次元を n とし, 残り半

分の 25 個のエージェントの予測式の次元を $n+1$ とした. 4000 期間のシミュレーションを行い, 各エージェントの予測値と市場価格との差を全期間で集積し, n 次元のエージェントグループと $n+1$ 次元のグループそれぞれについて平均をとった. そのようなシミュレーションを乱数の初期値を変えて 100 回行った^{†4}. 100 回の平均をとり, n 次元のグループの予測誤差を 100 として, $n+1$ 次元のグループの予測誤差がどれほど小さくなっているかを示したのが図 1 である.

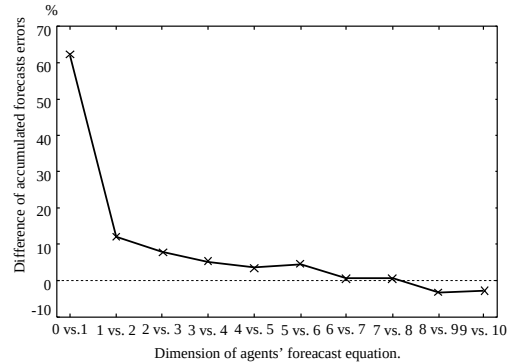


図 1 予測誤差の差: 値が正(負)の時は予測式を複雑にした方が予測誤差が小さく(大きく)なる.

次数が小さいうちは予測式を複雑にするメリットが大きい. 相対的により正確に予測できるエージェントが利得を増やせるので, 次数の小さいときは囚人のジレンマ状況の条件 1 が成り立っている. しかし, 次数が大きくなると, 予測式を複雑にするメリットがなくなることがわかる.

4.2 市場全体のデメリット

各エージェントが予測式の次元を上げると, 市場全体では予測しにくくなるのか確かめた. 50 個全てのエージェントの予測式を同じ n 次元にして, 4000 期間のシミュレーションを 100 回行った^{†5}. 予測誤差を 4000 期間で集積し, 50 個のエージェントの平均をとった後, さ

^{†4} 金融価格が発散してしまったパスは測定不能になるので, それは除外し発散しなかったパスで 100 回になるまでシミュレーションを行った

^{†5} 全てのエージェントの予測式が同じ次元の場合, 発散するパスは見られなかった.

^{†3} $n = 0$ の場合は, 乱数だけでの予測なので学習は行わない.

らに 100 回のシミュレーションの平均をとった (図 2).

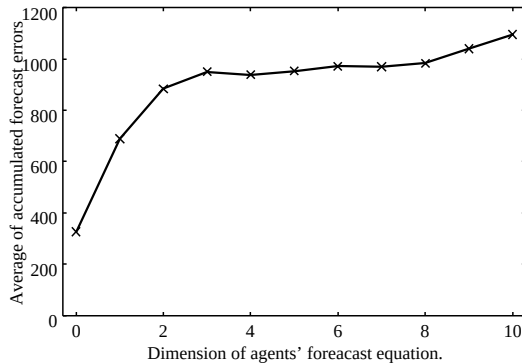


図 2 予測誤差

その結果, 次数が小さいときは, 次数の増加に伴い予測誤差が大きくなり, 囚人のジレンマ状況の条件 2 が成り立っていた. しかし, しかし次第に予測誤差がある一定の値に収束してしまったことがわかる.

4.3 市場の複雑さの発展

上記のメカニズムを調べるため, 相関次元分析を行った^{†6}. 50 個全てのエージェントの予測式を同じ n 次元にして, 4000 期間のシミュレーションを 100 回行った. 学習がある程度安定した後半の 3885 期間の市場価格の階差をとり, 埋め込み次元を変えながら, 相関次元を計算した (図 3).

その結果, 予測次元が 0 の場合 (図 3a) は, 平均は下に凸な曲線となり, ホワイトノイズに近い相関次元の特徴を持っていた. しかし予測次元が少し上がると (図 3b), 市場価格のダイナミクスに構造が現われて相関次元が飽和し上に凸な曲線になった. さらに予測次元を上げていくと (図 3c), ほぼ直線となり, 構造は存在するが相関次元が飽和しない状態になった.

中島 [8, 9] によると, 現実の東証株価指数データの相関次元分析結果も, 対数ベースで直線的に上昇した. つまり, 各エージェントの予測次元が増加していくと, 人工市場の価格のダイナミクスは, 現実の株価指数と同様に, 「いくつかの自由度をもつ力学系によって, (略) 挙動を大

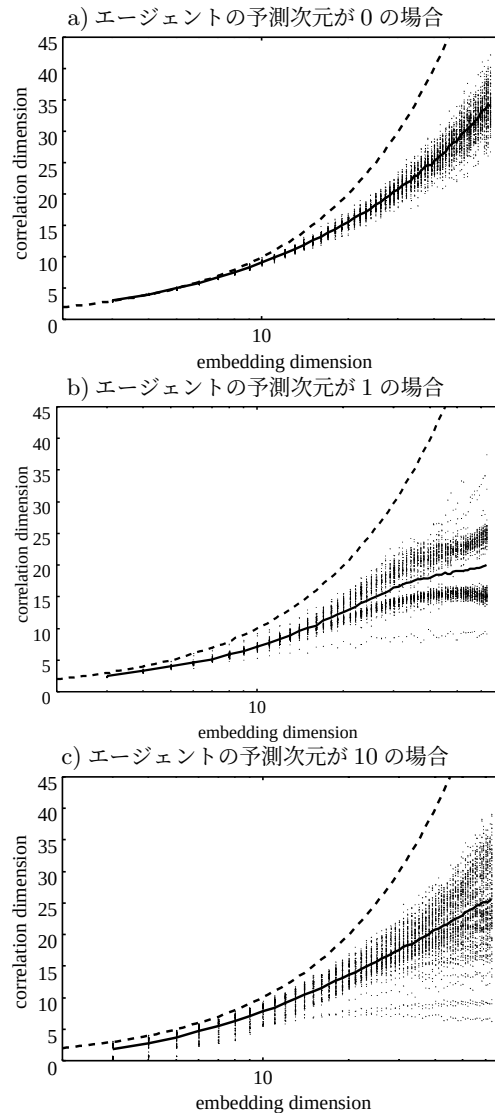


図 3 相関次元: x 軸 (埋め込み次元) は対数表示. 実線は 100 回のパスの相関次元の平均. 点線はホワイトノイズの場合の理論値.

まかに記述することができ, その記述はそれなりの有効性をもつ. そして, 自由度を増やすことによって, より精密な記述も可能となる. しかしながら, 完全にその挙動を再現する力学系は構成できない. すなわち, (略) 挙動に関わる変数の数は特定できない (中島 [9])」状態になるのである.

^{†6} 相関次元分析の手法は [8, 9] と同様に行なった.

5 新・効率的市場仮説

前節の結果をまとめると、下記のようなことがわかる。

- 各市場参加者の予測次元が 0 の時は、市場価格の挙動はランダムウォークであるが、予測次元を上げる人が出てくると価格の挙動にある自由度の力学系で記述できるような構造が出てくる。
- そのため、各エージェントが予測次元を上げると価格決定式の自由度に近づいて予測精度が上がるため、メリットが出てくる。しかし、皆が予測次元を上げると前よりも予測精度が下がり囚人のジレンマ的状况になる。
- 囚人のジレンマ的状况で皆が予測次元を上げていく方向に進んでいくと、市場価格の挙動の構造は存在するが、どの自由度でも完全に記述することができない状態になる。

今までの効率的市場仮説が言うような、各個人が学習により価格決定の構造を完全に把握し、その結果構造が消えてランダムウォークになることはなかった。市場の複雑さは個人の複雑さとともに変化し、最終的には構造は存在するが完全に把握することのできない状態になったのである。

このような学習の「正解」がないような状態では、学習の結果自体に加えて（よりも?）、各市場参加者の学習の仕方が市場価格の挙動に対する主要因になると考えられる。つまり、吉地 [7] が言うように、市場参加者の学習の効率性学習や市場を捕らえる認知的枠組みの違い、学習者間の相互作用、情報の取捨選択の仕方などが焦点になってくるのであろう。

また、もう 1 つの焦点として考えられるのは、市場価格決定の制度である。本研究では、価格決定はある一定の時間ごとに均衡価格を求める制度にしたが、連続的に取引が成立した価格を市場価格とする制度も考えられる。市場価格決定の制度は個人の複雑さをどのように市場の複雑さに集積するかというメカニズムであるので、市場の複雑さの発展の仕方に大きな影響を与える。制度を変えた場合に本研究と同じような最終結果になるのか、それとも異なるのかも興味深い。

6 結論

本研究で、人工市場アプローチを用いて効率的市場仮説の検討を行った結果、今までの効率的市場仮説と異なる以下の 2 点を見つけた。

- 市場の複雑さは個人の複雑さとともに変化する。個人の予測次元が小さいうちは、市場価格の挙動に記述可能な構造が存在し、囚人のジレンマ的状况により市場参加者が予測次元を上げる動機が存在する。しかし次元が上がっていくとその状況が壊れてしまう。
- 市場参加者が予測次元を上げていっても、市場価格の挙動の構造は消えない。最終的には構造は存在するが完全に把握することのできない状態になった。

今後の研究の発展の方向として、(a) 市場参加者の学習の仕方、(b) ファンダメンタルズ分析の取り込み、(c) 価格決定の制度などが市場の複雑さと個人の複雑さの関係にどのような影響を与えるか調べてみたい。

謝辞

本研究の遂行にあたり有益な議論や分析プログラムの提供をしてくれた中島義裕氏に深く感謝したい。

参考文献

- [1] Chen, S.-H. and Yeh, C.-H.: Genetic Programming and the Efficient Market Hypothesis, *Genetic Programming: Proceedings of the 1st Annual Conference*(Koza, J., Goldberg, D., and Fogel, D.(eds.)), the MIT Press, 1996, pp. 45-53.
- [2] Chen, S.-H., Yeh, C.-H., and Liao, C.-C.: Testing the Rational Expectations Hypothesis with the Agent-Based Model of Stock Markets, *Proceedings of International Conference on Artificial Intelligence 1999*, Computer Science Research, Education, and Application Press, 1999, pp. 381-387.
- [3] Chen, S.-H., Yeh, C.-H., and Liao, C.-C.: Testing the Rational Expectations Hypothesis with the Agent-Based Model of Stock Markets, 進化経済学論集第 4 集, 進化経済学会, 2000, pp. 142-145.
- [4] de la Maza, M.: Qualitative Properties of an Agent-based Financial Market Simulation, *Proceedings of ICAI99*, CSREA, 1999, pp. 367-373.
- [5] Harley, A. C.: *Time Series Models*, Philip Allan Publishers, 1981. (国友直人・山本拓訳『時系列モデル入門』東京大学出版会, 1985.).
- [6] Joshi, S., Parket, J., and Bedau, M. A.: Technical Trading Creates a Prisoner's Dilemma: Results from an Agent-Based Model, *Computational Fi-*

- nance 1999*(Abu-Mostafa, Y. S., LeBaron, B., Lo, A. W., and Weigend, A. S.(eds.)), MIT Press, 2000, pp. 465–479.
- [7] 吉地望: 不確実性下での期待形成と仮説の進化—為替相場における期待形成を通して—, 方法としての進化 (塩沢由典 (編)), シュプリンガー・フェアラーク東京, 2000, chapter 6, pp. 173–206.
- [8] 中島義裕: 経済現象に見られる決定論的性質と確立論的性質の両義性, 情報処理学会論文誌:『数理モデル化と応用』, Vol. 40,No. SIG9(TOM2)(1999).
- [9] 中島義裕: 経済の揺らぎとフラクタル, 方法としての進化 (塩沢由典 (編)), シュプリンガー・フェアラーク東京, 2000, chapter 7, pp. 207–235.

X-Economy – マルチエージェント 経済におけるシミュレーションプラットフォーム –

川村 秀憲, 車谷 浩一, 大内 東

現在様々なスケールの経済活動の根本的なメカニズムを解明するためにマルチエージェント経済学が盛んになりつつある。しかし、マルチエージェント経済学はその方法論上の幾つかの問題点がある。それらは大きく分けて、モデルの記述・表現に関する問題、構築・評価に関する問題、モデルの共通化・蓄積に関する問題の3つの問題に代表される。そこで我々の研究グループでは、それらの問題点を明らかにするとともに、それらの問題を乗り越えてマルチエージェント経済研究を発展させるために、X-Economyプロジェクトを立ち上げ、マルチエージェント経済のためのシステム提供と実際にそれを用いた研究プログラムを展開している。ここではX-Economyプロジェクトの概要、シミュレーションプラットフォームの紹介と幾つかの研究事例について紹介する。

1 はじめに

近年の複雑系研究の発展に伴って、複雑性を持った主体の行動をエージェントとしてモデル化するエージェントモデリングと、計算機シミュレーションによってエージェントのミクロ・マクロな挙動を解析するエージェントベースシミュレーション研究が盛んになりつつある。

これらの手法によって、学習や進化によってマクロな挙動からミクロな行動ルールを変化させるような行動主体をエージェントとしてモデル化することが可能になり、従来は対象として扱うのが困難であった領域にもモデル化とシミュレーション実験の可能性を広げている。

これらの状況から、仮説に対する検証実験を行うのが困難である社会科学でもエージェントベースシミュレーションが盛んになりつつある[1]。その中でも、アーサー等による人工市場研究に代表されるような経済現象に関するエージェントベースシミュレーションが注目を集めている[2][3]。エージェントベースシミュレーションは、従来ほぼ不可能であった経済現象に対する実験のパラダイムとして有効であるばかりではなく、単なる仮説の検証から制度設計論への発展まで含めて経済学での重要性を増していくように思われる[4]。

しかし、エージェントベースシミュレーションには3つの大きな問題がある[5][6][7]。一つ目はモデルの記述・表現に関する問題である。通常エージェントモデルは計算機上にプログラムとして作成されるため、柔軟性・表現力は大きい。しかし、学習や進化を含めたエージェントの意思決定は複雑に構築されることが多く、またモデル自体がシミュレーションプログラムに組み込まれてしまっているために、プログラムからモデルを切り離して記述・表現することが困難である。

二つ目はモデルの構築・評価に関する問題である。エージェントベースシミュレーションが扱う対象は基本的に複雑な世界であり、モデリングが対象の性質をうまく捉えて妥当に構築されているのかどうかを判断するのは困難である。また、得られる結果と対象とをどのように比

X-Economy - Simulation Platform for Multi-agent Economics

Hidenori Kawamura, 北海道大学大学院工学研究科, Graduate School of Engineering, Hokkaido University.

Koichi Kurumatani, 産業技術総合研究所サイバーアシスト研究センター, Cyber Asist Research Center, National Institute of Advanced Industrial Science and Technology.

Azuma Ohuchi, 北海道大学大学院工学研究科, Graduate School of Engineering, Hokkaido University.

較していけばよいのか、そして第三者がその結果をどう客観的に解釈すればよいのかといったことに関する統一的な方針が無いという問題もある。

三つ目は、モデルの共通化・蓄積に関する問題である。これは、上の二つの問題があるために、それぞれの研究者が各自の目的に合わせてオリジナルのモデルを作成することが多く、過去の研究成果を有効に共有・蓄積するのが困難であるという問題である。これらの問題を解決していかなければ、エージェントモデリングが有効な研究上のパラダイムとして確立出来ないのは明らかである。

そこで、これらの問題点の一つの解決策として、我々はエージェントベース経済学のための *X-Economy* プロジェクトを展開している [8][9]。ここでは、WWW 上に構築された *X-Economy* プロジェクトのフォーラムを中心に、人工市場に基づくエージェントベースシミュレーションシステムである *X-Economy* システムの提供と、それを用いたリサーチプログラムの共有を目指している。つまり、マルチエージェント経済における Linux と呼べるようなシステムの開発と研究スタイルの提案を目的としている。各研究者がシステムを共有し、用いたモデルとシステムを公開・提供する場所を用意することで、先ほど上げた問題点の一つの解決策になると考えている。これらの状況を踏まえ、本稿では我々が試験的に構築した *X-Economy* システムのプロトタイプの概要を紹介する。

2 人工市場研究の為のエージェントベースシミュレータ

まず始めに既存のエージェントベースシミュレータについて述べる。現在、人工市場研究に関わらず幾つかの有償・無償のエージェントベースシミュレータが提供されている。代表的なものでは、Santa Fe 研究所が提供する Swarm シミュレータや MIT の StarLogo、構造計画研究所の ABS などがある。これらのエージェントベースシミュレータは、エージェントのモデリングが直感的に行えたり、グラフィカルに振る舞いやデータを表示できる機能を持っているなど、シミュレーションを行う上で非常に利便性が高い。事実 Swarm に基づく人工市場モデルも公開されている。

これらのシミュレータを用いて行うエージェント研究

は、シミュレーションの実行エンジンとエージェントの記述が分離されているために、エージェントモデルの記述には優れているが、モデル自体がシミュレータに特化していたり、またシミュレータの詳細な動作がユーザに隠蔽されていたりする場合がある。また、シミュレータの自由度が高すぎて、計算機プログラムを設計するのと同じぐらいの労力がかかる場合もある。

そこで我々のアプローチは、ある程度人工市場研究に必要なと考えられる幾つかの要素をモジュール化してプログラムを設計し、そのプログラムのソースコードを元にユーザが必要な機能を追加することで、シミュレーションシステムの提供を試みる。また、ユーザが作成した新たな機能モジュールをライブラリーとして蓄え再公開することによって、モデルをプログラムとして表現し、また他のユーザの研究成果の相互参照を容易にすることが可能になる。シミュレータの提供と言うよりむしろ、人工市場研究のためのシミュレーションプログラムのテンプレートを提供するという感覚に近い。この様な取り組みは、Robocup におけるサッカーエージェントのシミュレーション部門でも行われており、エージェント研究の進展に大きな貢献をしている [10]。

ここで重要なのは、モジュール化の概念に基づくテンプレートの設計である。人工市場研究では、シミュレーションの要素として多種・複数のトレーダエージェントとマーケットの動作が基本となっている。そこで *X-Economy* システムでは、それぞれを個別のプログラムとして設計し、各プログラムの入出力を X-SS(eXtensible Social System) プロトコルのサブクラスである MAEP (Multi-Agent Artificial Economy Protocol) プロトコルとして定義する事によって、モジュール間の相互作用を定義している [11][12][13]。またそれらの通信はテンプレートプログラムに埋め込まれているので、ユーザはモジュール間の通信を意識することなく *X-Economy* システムの新しい機能を設計し、望みのエージェントモデルを設計することが出来る。システムの概要を次に説明する。

3 *X-Economy* システム

ある程度の汎用性をもったマルチエージェント経済シミュレーションのためのシミュレーションプラットフォームを作成するにあたり、*X-Economy* システムでは大ま

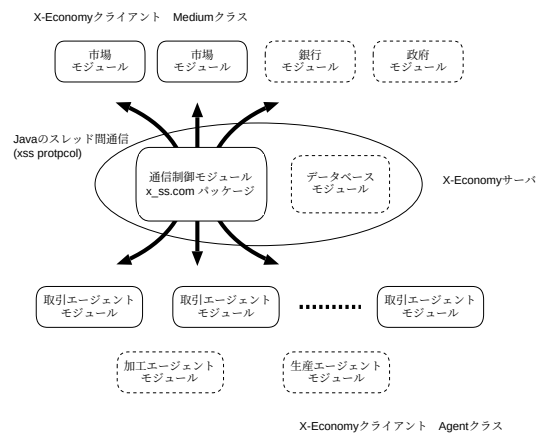


図1 X-Economyシステムの概念図（点線で囲まれた部分は現在未実装）

かにいって以下のシステムの要素を考えている。

- X-Economyサーバ：このサーバはシステム間の複数のモジュールの通信を制御する。このサーバを介して複数のモジュールが通信し、システム全体の複数の構成要素がそれぞれ役割を担いながらシミュレーションが進行する。
- Medium クラスモジュール：具体的には人工市場モジュール（市場型取引・相対型取引）、銀行モジュール、政府モジュールなど経済システムにおいて、メタな役割を担うエージェントを実現するモジュールである。
- Player クラスモジュール：具体的には、市場に対して取引を行う取引エージェントや、農作物などの生産する生産エージェント、また生産物を加工するための加工エージェント等である。

図1はこれらの要素からなる X-Economyシステムの概念図である。複数のモジュールが X-Economyサーバを介して通信し、シミュレーションが進行する。

現在このような概念に基づく X-EconomyシステムをJavaを用いて開発しており、Medium クラスモジュールとして市場モジュール、Player クラスモジュールとして取引エージェントモジュールが完成している。取引エージェントは複数の銘柄の財を持ち、それぞれの銘柄の財の交換注文を市場モジュールにサーバを介して送る。市場モジュールは各取引エージェントの注文を集め、そ

れらを取りまとめることによって財の交換の仲立ちをする。この二つの要素からなるシミュレーションとしては、株式などの変動リスク資本と通貨などの無リスク資本に基づく人工市場シミュレーションなどが出来ることになる。他の要素は今後順次実装していく予定であるが、モジュール間の通信などシステムの基本的な部分はすでに完成しているので、拡張は容易である。

具体的にシステムの実装についてであるが、現在はそれぞれのモジュールをJavaのスレッドとして実現している。また、モジュール間の通信はスレッド間の通信として定義されているが、将来的に複数のプロセス・複数のプラットフォームでの分散シミュレーション環境を想定し、通信プロトコルはXMLベースで開発を進めている。具体的に、X-EconomyプロジェクトではX-SSプロトコルのサブクラスのMAEPプロトコルとしてモジュール間の通信プロトコルを定義している。

MAEPプロトコルの内容であるが、一般にシステムの構成要素として n 種類のモジュールが存在する場合、 $n(n-1)/2$ 種類のプロトコルが必要である。また、新たなモジュールの追加によって n 種類のプロトコルを追加しなければならず、組み合わせ的にどんどん実装が困難になっていく。そこでMAEPプロトコルではこの点を解決するために、システムに流通する「財」や「情報」をオブジェクトと見なし、「オブジェクトの交換」という形で取引や情報の流通を捉え、交換されるオブジェクトの種類に応じてプロトコルを定義する。例えば取引エージェントが市場モジュールに出す取引要求のプロトコルは、XMLのDTDとして図2の様に定義される。

このように、モジュール間の通信プロトコルをXMLとして定義することによって、将来的に複数のプラットフォームでの分散型シミュレーションや、シミュレーション結果の解析、システムの相互利用、そしてデータ解析手法の共通化などが可能となる。現在はこのプロトコルを拡張していきながら、システムの開発を進めている。

4 X-Economyシステムを用いた研究事例

現在我々のプロジェクトでは X-Economyシステムを用いた幾つかの研究を行っており、ここでは、X-Economyシステムを用いた人工市場研究の事例を紹介する。これらの研究では、基本的に X-Economyシステムの幾つか

```

<!--
goods-trade-request.dtd (general agent -> medium, and v.v.)
Copyright Mar. 8, 2001.
Koichi Kurumatani, Cyber Assist Research Center.
All rights reserved.
-->

<!ELEMENT goods-trade-request (order, item, quality, country, market-name, value, pastmsgid)>
<!ELEMENT order      (#PCDATA)>          <!-- buy, sell, cancel, inquiry -->
<!ELEMENT item       (#PCDATA)>          <!-- wheat, flour, bread, steel, machine, house -->
<!ELEMENT quality    (#PCDATA)>          <!-- reserved -->
<!ELEMENT country    (#PCDATA)>          <!-- name of country (GID) -->
<!ELEMENT market-name (#PCDATA)>          <!-- name of market (UID) -->
<!ELEMENT client-name (#PCDATA)>          <!-- name of client (UID) -->
<!ELEMENT value      (price-range, quantity-range)>
<!ELEMENT price-range (low, high)>
<!ELEMENT quantity-range (low, high)>
<!ELEMENT low        (#PCDATA)>          <!-- long integer -->
<!ELEMENT high       (#PCDATA)>          <!-- long integer -->
<!ELEMENT pastmsgid  (#PCDATA)>          <!-- long integer -->

<!-- About "pastmsgid"
- From agent to medium:
This field is valid only when "order" is "cancel".
The message id. which you want to cancell should be assigned to "pastmsgid."
-->

<!-- end of goods-trade-request.dtd -->

```

図2 MAEPのDTD定義の一例[12][13].

のモジュールの機能部分を追加・変更したシステムが用いられている。

- **ニュース情報に基づく取引エージェントを用いた人工市場に関する研究 (北海道大学：兼平大輔 他)**

従来の人工市場におけるトレーダエージェントは、過去の値変動を参考に意思決定するものがほとんどであり、外部的に与えられる情報を利用したトレーダエージェントの構築と、その情報の影響に関する研究はまだあまり為されていない。従って、人工市場研究の発展の為にそれらを扱っていくエージェント技術の発展と研究の方法論の確立も必要である。そこで兼平等の研究では、ある企業の現実のニュースに着目して、そのニュースの情報をDBに蓄えて人工市場上のトレーダエージェントに配信し、ニュース情報が市場の振る舞いに与える影響について研究を行っている[14]。

- **国際取引における人工市場モデルに基づく基軸通貨の創発に関する研究 (北海道大学：佐々木雄一 他)**

基軸通貨とはドルのように複数の国にまたがる取引において広く用いられる通貨のことであり、国際経済のような複雑なシステムの理解の為に重要な要素の一つである。基軸通貨が創発するメカニズムとして、通貨発行国が(1)資本の提供国であること、(2)大規模な経済を持っていること、等が挙げられているが、現実にそのメカニズムは詳しくわかっていない。従って、国際経済と基軸通貨の関係を調べることは、国際貿易政策や制度設計の意味も含めて、重要な課題である。佐々木等の研究では基軸通貨の創発の基本的なメカニズムについて調べるために、シンプルな国際貿易エージェントモデルに基づくシミュレーションを行っている。

- **複数投資対象への分散投資エージェントの設計と市場の挙動に関する研究 (北海道大学：山本隆人 他)**

現在人工市場研究で扱われている投資対象は、抽象化された単一のリスク資本に対して行われることがほとんどである。しかし現実の市場では金融工学・

ポートフォリオ理論に基づく分散投資が当たり前であり、それに基づいて複数の投資対象間の相互作用が起こっていると捉えるのが自然である。そこで山本等の研究では、実際に複数投資対象にポートフォリオ理論に基づいて分散投資を行うエージェントをモデル化し、分散投資によって各投資対象間にどのような相互作用が総じているのかを明らかにするために、*X-Economy*システムを用いて研究を行っている [15].

● エージェント実験経済標準モデルとしての国際貿易リーグの提案 (産業総合技術研究所：車谷浩一)

上の佐々木の研究例にもあるように、近年一国・多国の経済システムをマルチエージェントシステムとしてモデル化し、それらの相互作用とそこから創発するマクロな振る舞いの間に潜むメカニズムを解明しようという研究がなされつつある。しかし一般に国際貿易を含めたエージェントモデルの開発、またはそれを用いた研究というのは、対象があまりに複雑であるので、モデルの抽象度やシミュレーションの方法、実験方法、結果の検証まで含めて様々な研究者が相互に成果交換を行いながら研究を進めていかなければ有効な研究の成果となり難い。また、そのような複雑なシステムでのエージェントの意思決定を議論する上では、各研究者でモデルに対する共通認識が重要になってくる。このような観点から、車谷は *X-Economy* システムを用いたエージェント実験経済標準モデルとしての国際貿易リーグの提案を行っている。

5 おわりに

本稿では、我々が現在取り組んでいる *X-Economy* システムの概要と、それを使った幾つかの人工市場研究について紹介した。特にこれらの研究は、*X-Economy* システムを用いることで、比較的容易にシミュレーションが行えると同時に、実験プログラムを公開するという前提に立つことによって、エージェントモデリングにおけるシミュレーションモデルの記述という観点からも有効である。

エージェントベースシミュレーションが真に意味のある方法論になる為の次のステップとして、エージェント

モデルをどうやって記述・蓄積していくかが大きな課題であるが、*X-Economy* システムのような取り組みは一つの試金石になるだろう。

エージェントベースシミュレーションによって複雑な現象を解明するためには、本プロジェクトのように研究者間での知識・方法論の共有化が重要であり、興味をお持ちの方々とはぜひ協力できればと考えている。

なお、ここで紹介したJavaによる *X-Economy* システムはプロトタイプであり、現在公開への準備を進めている。

本研究は、文部科学省科学研究費補助金1378034によって行われました。

参考文献

- [1] 寺野隆雄, 倉橋節也: “エージェントシミュレーションと人工社会・人工経済”, 人工知能学会誌, Vol. 15, No. 6, pp. 966-973 (2000).
- [2] Arthur B., Holland H., LeBaron B., Palmer R., Tayler P.: “Asset Pricing Under Endogenous Expectations in an Artificial Stock Market”, Santa Fe Working paper (1996).
- [3] 和泉潔, 植田一博: “人工市場入門”, 人工知能学会誌, Vol. 15, No. 6, pp. 941-950 (2000).
- [4] 塩沢由典: “経済学にとっての人工市場”, 人工知能学会誌, Vol. 15, No. 6, pp. 951-957 (2000).
- [5] Moss S.: “Editorial Introduction: Messy Systems - The Target for Multi Agent Based Simulation”, Multi-Agent-Based Simulation, LNAI 1979, Springer, pp. 1-14 (2000).
- [6] Edmonds B.: “The Use of Models - Making MABS More Informative”, Multi-Agent-Based Simulation, LNAI 1979, Springer, pp. 15-32 (2000).
- [7] Takadama K., Shimohara K.: “Toward Cumulative Progress in Agent-Based Simulation”, JSAI 2001 International Workshop on Agent-based Approaches in Economic and Social Complex Systems (AESCS 2001), pp. 35-42 (2001).
- [8] *X-Economy* プロジェクト Web Site, <http://www.x-econ.org/>
- [9] 車谷浩一, 大内東: “国際貿易リーグ: マルチエージェント経済における標準問題(1) - 概念と *X-Economy* システムによる実現 -”, 情報研報, Vol. 2001, No. 1, pp. 55-60 (2001).
- [10] Robocup 公式 Web Site, <http://www.robocup.org/>
- [11] Kurumatani K., Ohuchi A., World Trade League as a Standard Problem for Multi-Agent Economics - Concept and Background, Proceedings of International Workshop on Agent-based Approach to Economic and Social Complex Systems, AESCS'01, pp. 156-162 (2001).
- [12] 幸島明男, 車谷浩一: 人工経済用エージェント間通信

プロトコル MARP, 情報処理学会論文誌投稿中.

[13] X-SS Web Site, <http://www.x-ss.org>

[14] 兼平大輔, 川村秀憲, 大内東, 車谷浩一: “*X-Economy* システムを用いた人工株式市場におけるニュースの情報価値に関する研究”, 2001年度日本オペレーションズ・リサーチ学会春期研究発表会アブストラクト集, pp. 252-253 (2001).

[15] 佐々木雄一, 山本雅人, 大内東, 車谷浩一: “*X-Economy* システムを用いた人工国際市場における基軸通貨の発生に関する研究”, 2001年度日本オペレーションズ・リサーチ学会

春期研究発表会アブストラクト集, pp. 254-255 (2001).

[16] 川村秀憲, 大内東, 車谷浩一: “人工市場モデルのための *X-Economy* システムとその利用研究”, 日本オペレーションズ・リサーチ学会学会誌, Vol. 46, No. 10, pp. 537-543 (2001).

[17] 車谷浩一: “国際貿易リーグ: マルチエージェント経済における標準問題 - 概念と *X-Economy* システムによる実現”, 日本オペレーションズ・リサーチ学会学会誌, Vol. 46, No. 10, pp. 544-548 (2001).

P2P 型電子商取引のための エージェントの提携に基づく売買交渉機構について

服部 宏充 伊藤 孝行 新谷 虎松

1 はじめに

近年、電子商取引 (EC: Electronic Commerce) が社会的にも高い関心を集めている。電子商取引は、インターネット上で行われる売買活動であるため、実世界で行われるものとは異なるタイプの、効果的な商取引が実現できる可能性がある。現在、人工知能の分野では、電子商取引を対象とした研究が活発に行われている。特に注目を集めているアプローチとして、エージェントに基づく電子商取引支援がある [10]。エージェントとは、自律的かつ協調的に振る舞い、人間の代理として様々なタスクを実行可能なソフトウェアであり、計算機上での知的な活動主体である。ユーザは、エージェントを用いることによって、希望する商品を探し出したり、実際に交渉する負担から解放される利点が得られる。エージェントが主体となって売買交渉を行うことにより、インターネット上で、ソフトウェアを中心とした新たな経済活動のパラダイムが生み出されることが期待される。ここでは、エージェン

トが、売買に関する多くの情報を知的に処理し、効果的な売買を自律的に行うことが求められる。ゆえに、エージェントを中心とした新奇な売買メカニズムの開発が必須である。

エージェントに基づく電子商取引支援に関しては、これまでに様々な研究が行われている。筆者らは、これら既存の研究を 2 つのカテゴリに分類している。一方は、エージェントに基づく電子マーケットに関する研究で、Kasbah [4], Tete-A-Tete [9], FishMarket [15], および AucutionBot [18] などの研究がある。もう一方は、売買に関する情報収集エージェントに関する研究がある [5][6][12]。現在筆者らは、前者の、エージェントに基づく電子マーケットに関する研究に注目している。

本論文では、実際に複数のエージェントが売買交渉を行う場として、電子マーケット *G-Commerce* を提案する。既存の多くのシステムは、実世界における市場のメタファをネットワーク上のソフトウェアとして実現した、中央集中型のシステムである。そのため、高度に非同期分散処理が可能なインターネットの利点を十分に活かしていない。一方、*G-Commerce* は Peer-to-Peer(P2P) [14] に基づく分散型のシステムである。P2P とは、Napster [13] や Gnutella [8] に代表されるネットワークモデルのことである。*G-Commerce* では、インターネット自体を、複数の買い手/売り手エージェントが存在する 1 つの巨大な市場とみなす。その中で、ある商品の売買に関係する複数のエージェントが、一時的な (ad-hoc な) 売買ネットワークを形成し、各エージェントは特定のサーバを介

On a Negotiation Mechanism based on Coalition Formation among Agents in a P2P Electronic Marketplace.

Hiromitsu HATTORI, 名古屋工業大学大学院 工学研究科, Graduate School of Engineering, Nagoya Institute of Technology.

Takayuki ITO, 北陸先端科学技術大学院大学 知識科学教育研究センター, Center for Knowledge Science, Japan Advanced Institute of Science and Technology.

Toramatsu SHINTANI, 名古屋工業大学 知能情報システム学科, Dept. of Intelligence and Computer Science, Nagoya Institute of Technology.

さずに直接売買交渉を行う。現在、P2P の形式には、Hybrid P2P 型と Pure P2P 型の 2 つのタイプがあると考えられている。Hybrid P2P 型は Napster が採用している形式で、各ユーザが持つ情報を管理するサーバを有しており、従来のサーバ／クライアント形式を一部取り入れたものとなっている。一方、Pure P2P 型は Gnutella が採用している形式で、Hybrid 型のようなサーバを持たない。完全な分散型システムの実現のためには、後者の Pure P2P 型が望ましいが、各ユーザが持つ情報の把握が困難である点や、最初に誰と接続するのか、というエントリポイントに関する問題が存在する。筆者らは、両者を組み合わせたシステムアーキテクチャの構築を試みている。本論文では特に、*G-Commerce* において、エージェントが適切な売買交渉の相手を発見するための手法を示す。

G-Commerce には、買い手、もしくは売り手となる複数のエージェントが存在することになる。既存のシステム (e.g., Kasbah, Tete-A-Tete) では、売買交渉は 1 つの買い手エージェントと 1 つの売り手エージェントによって行われていた。すなわち、複数の買い手／売り手エージェントがマーケット内に存在するにもかかわらず、各々の売買交渉は互いに独立して行われていた。筆者らは、売買交渉の際に、複数の買い手／売り手エージェントが相互に関係する場合を考慮した交渉手法の開発を試みている。文献 [19] では、交渉プロトコルの一種であるオークションにおいて、複数の買い手エージェントが提携する仕組みが提案されている。本論文では、売り手エージェントと買い手エージェントの双方が提携可能とした場合での、売り手エージェントの提携手法について明らかにする。具体的には、売り手エージェントは、交換に基づく協調的な交渉 [11] によって提携を形成する。

本論文の構成は以下の通りである。第 2 章では、*G-Commerce* の概要を示し、システムにおけるエージェントのマッチング手法について述べ、語彙の定義と交渉のプロトコルを示す。第 3 章では、売り手エージェント間の、交換に基づく交渉手法について論じる。第 4 章では、提案した交換に基づく交渉手法について、理論的／実験的な評価を行い、システムの実行例を示す。第 5 章では関連研究を示し、本研究との

相違点を明らかにする。第 6 章で結論を述べる。

2 *G-Commerce*: P2P に基づく電子マーケット

2.1 *G-Commerce* の概要

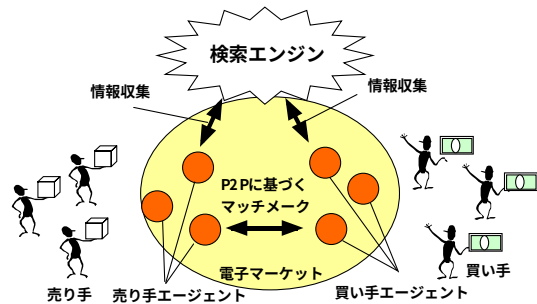


図 1 Outline of *G-Commerce*

図 1 に、P2P に基づく電子マーケット *G-Commerce* の概要を示す。*G-Commerce* には、ユーザの代理となるエージェントが複数存在し、自律的かつ協調的に商取引を行う。ユーザである売り手と買い手は、それぞれ、売り手エージェントおよび買い手エージェントを生成する。売り手エージェントと買い手エージェントは、お互いに適切な相手を探しだし、交渉を行う。

G-Commerce は P2P に基づくシステムであり、市場における情報管理のための特別なサーバを必要としない。ユーザは、個々の計算機において、エージェントが動作する場を提供するためのソフトウェアを稼働させるだけで良い。各ユーザの計算機上には、ユーザの好み (嗜好) を保持し、売買交渉を代行するエージェントが、少なくとも 1 つ存在している。エージェントは、モバイルエージェントとして実装されており、適切な交渉相手を発見した後、ネットワークに接続された他の計算機に移動して、直接売買交渉を行うことができる。また、エージェントは WWW サーバとしての機能を持ち、HTTP アクセスに対して、適切な内容の HTML を生成する CGI プログラムとして動作する。従って、エージェントは、入力されたユーザの好みに適切な処理を与え、通常の Web ページと同様の形式でインターネット上に公開することができる。ここでエージェントが公開する情報は、

交渉相手として適切なエージェントのマッチングを実現するために利用される。エージェントのマッチングについては、第 2.2 節で述べる。

本システムの実装には、*MiLog*(Mobile intelligent agents using Logic programming) [7] を用いている。*MiLog* は、Java によって実装された知的エージェント開発環境であり、Prolog と Java を用いて、知的なエージェントをプログラミングすることが可能である。*MiLog* では、強モビリティ、WWW サービス機能、および WWW アクセス機能を持つエージェントを生成できる。すなわち、*MiLog* のエージェントは、CGI プログラムや Servlet のようにサーバーサイドプログラムとして振る舞うと同時に、HTTP を用いて他の WWW サーバーにアクセスすることが可能である。

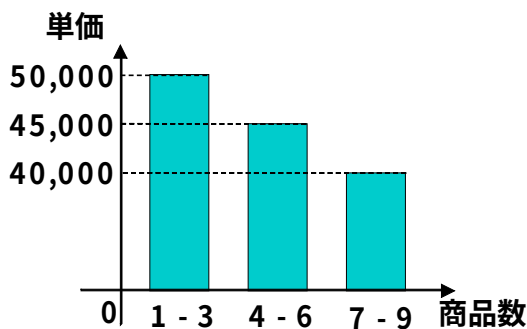


図 2 売り手の価格テーブルの例

G-Commerce では、1 つの売り手エージェントが、複数の商品を販売することができる。各商品には、価格テーブル、締め切り、および在庫数が定められている。価格テーブルとは、商品の販売数に基づく割引価格を示すテーブルであり、締め切りとは、その商品を売ることができる期限である。図 2 に、価格テーブルの例を示す。本例では、買い手エージェントが 5 つの商品を買った時、商品の単価が 45,000 円になることが示されている。

2.2 エージェントのマッチングメイキング

本節では、*G-Commerce* における、エージェントのマッチング手法を明らかにする。具体的なマッチン

グのプロセスを以下に示す。

【Process 1】

ユーザは、エージェントに対して、売買に関する希望 (e.g., 購入, もしくは販売希望商品に関する選好) を入力する。現在、本システムではユーザの選好として、商品のカテゴリ、商品名、希望価格等を用いている。

【Process 2】

第 2.1 節で述べた通り、エージェントは WWW サーバの機能を持つため、通常の Web ページと同様の HTML による記述に基づいてユーザの選好をインターネット上に表明することができる。買い手エージェントは、ユーザの商品に関する選好を表明する HTML 文書に、特殊なキーワード *UniqueKeyword*, および各々のサーバへのアクセス方法 (e.g., IP アドレス) 等をコメントとして埋め込む。

【Process 3】

Google, goo 等のロボット型検索エンジンでは、検索ロボットがネットワーク上を自動巡回し、Web データを収集する。*G-Commerce* のエージェントも、通常の Web ページと同様、検索ロボットによるアクセスを受け付けることが可能である。検索ロボットは、エージェントが生成した、ユーザの選好とキーワード *UniqueKeyword* 等を含む HTML データを収集し、各検索エンジンのデータベースに記録する。ここで、エージェントは、保持しているユーザの選好に基づいて、表明する情報を適切に調整する。

【Process 4】

適切な交渉相手を探すために、エージェントは、検索エンジンを利用したエージェントサーチを行う。最もシンプルな方法として、検索エンジンに対して、商品名とキーワード *UniqueKeyword* との “AND” 検索を行う。例えば、商品 *Item* の販売を希望しているエージェントが、買い手のエージェントを探す場合は、*Item* と *UniqueKeyword* との AND 検索を行う。ここでの検索結果は、本システムを利用しており、かつ商品 *Item* の購入を希望している買い手エージェント

のリストとなる。リストはエージェント内部にキャッシュされ、次回以降、同様の商品の売買を行う際に利用される。

【Process 5】

エージェントは、検索の結果得られたリストからサーバへのアクセス方法を抽出し、各々のエージェントの計算機上へ移動し、売買交渉を開始する。

本手法の特長は、他のエージェントの情報を参照するためのサーバとして、通常は Web ページを検索するために用いる検索エンジンを利用している点である。そのため、P2P の形式は Hybrid P2P 型と同様になるが、ユーザの側でサーバを用意する必要はない。エージェントが複数の検索エンジンを同時に用いることにより、インターネット上に存在するほぼ全てのエージェントの発見が可能となる。また、同種の商品に関する売買を繰り返し行うならば、エージェントが、過去に交渉を行ったエージェントの情報を内部にキャッシュしておくことにより、本手法を用いることなく、直接売買交渉を行うことができる。すなわち、Pure P2P 型と同様の動作が可能となる。

2.3 *G-Commerce* におけるエージェント間の交渉プロトコル

まず、本論文で用いる語彙を以下に定義する。

買い手エージェント: $B = \{b_1, b_2, \dots, b_l\}$ は、買い手エージェントの集合を表す。

売り手エージェント: $S = \{s_1, s_2, \dots, s_m\}$ は、売り手エージェントの集合を表す。

商品: $G = \{g_1, g_2, \dots, g_n\}$ は、商品の集合を表す。特に、売り手 s_j の持つ商品の集合を、 $G(s_j) = \{g(s_j)_1, g(s_j)_2, \dots, g(s_j)_n\}$ と表す。売り手は、複数の種類の商品を売ることが可能とする。

買い手の希望商品: $g(b_k)$ は、買い手 b_k の希望する商品を表す。

価格テーブル: 売り手 s_j の商品 g_i に対する価格テーブルは、関数 $p(s_j, g_i) : N \rightarrow R$ で表される。こ

こで、 N は自然数、 R は実数である。 $p_n(s_j, g_i)$ は、商品 g_i が n 個売れた場合の g_i の単価を示す。

商品のコスト: $cost(s_j, g_i)$ は、売り手 s_j が商品 g_i を仕入れるため、または生産するためにかけるコストを表す。

希望価格: $r(b_k, g_i)$ は、買い手 b_k の商品 g_i に対する希望価格を表す。

売り手が提示する商品数: $num(s_j, g_i)$ は、交換に基づく交渉において、売り手 s_j が提示する商品 g_i の数である。

買い手が希望する商品数: $num(b_k, g_i)$ は、買い手 b_k の商品 g_i に対する希望個数を表す。

買い手の提携: $C(g_i)$ は、商品 g_i に対する買い手の提携を表す。ここで、 $C(g_i) \subset B$ である。

提携としての商品の希望個数: $num(C(g_i)) = \sum_{b_k \in C(g_i)} num(b_k, g_i)$ は、提携 $C(g_i)$ が希望する商品 g_i の個数を表す。

買い手の効用: 買い手 b_k が、売り手 s_j から商品 g_i を買った時、 b_k の効用は以下のように定義される。 $u_{b_k}(g_i) = r(b_k, g_i) \times num(b_k, g_i) - p_{num(b_k, g_i)}(s_j, g_i) \times num(b_k, g_i)$ 。

買い手の提携の効用: 提携 $C(g_i)$ の効用は以下のように定義される。 $u_{C(g_i)}(g_i) = \sum_{b_k \in C(g_i)} r(b_k, g_i) \times num(b_k, g_i) - p_{num(C(g_i))}(s_j, g_i) \times num(C(g_i))$ 。

G-Commerce では、以下のプロトコルに従ってエージェントが売買交渉を行う。ここでの基本的なアイデアは、単独での商品の売却に失敗した売り手エージェントが、他の売り手エージェントと在庫にある商品を交換することにより、買い手エージェント、もしくは買い手エージェントの提携が希望する商品数を集めて商品の売却を試みる、というものである。

(Step 1): 買い手エージェント b_k は、売買に関する要求（商品の種類 g_i 、希望価格 $r(b_k, g_i)$ 、および個数 $num(b_k, g_i)$ ）を宣言する。第 2.2 節で述べた方

法により、同じ種類の商品 g_i を希望する複数のエージェントにより、提携 $C(g_i)$ が形成される。

(Step 2): 売り手エージェント s_j が商品 g_i に関する提携 $C(g_i)$ を発見し、売り手エージェント s_j の持つ商品の中に商品 g_i が含まれている、すなわち、 $g_i \in G(s_j)$ ならば、商品 g_i の売却を試みる。もし、商品 g_i に対する提携 $C(g_i)$ が存在しなければ、自分が持っている商品の集合 $G(s_j)$ に含まれる商品に対する、買い手エージェントの提携が形成されるのを待つ。ただし、待つのは、ユーザに与えられた締め切り時間までである。

(Step 3): $\text{num}(s_j, g_i) \geq \text{num}(C(g_i))$ が成り立つなら、売り手エージェント s_j は、商品 g_i の提携 $C(g_i)$ への売却に成功し、交渉は成立する。もし、 $\text{num}(s_j, g_i) < \text{num}(C(g_i))$ ならば、(Step 4) へ。

(Step 4): 買い手の要求を満足させるために、売り手エージェント s_j は、交換に基づく交渉手法を用いて、商品 g_i の数を増やすことを試みる。例えば、売り手エージェント s_j が、他の売り手エージェント s_k から、より多くの商品 g_i を得るために、 s_j は g_i の代償として、その他の商品を s_k に渡す。交換に基づく交渉の詳細は、第 3 章で述べる。

(Step 4) において、もし、商品を交換可能なエージェントが複数存在する場合、売り手エージェント s_j は効用を最大化するエージェントと交換を行う。もし、商品 g_i の交換が可能なエージェントが存在しない場合、売り手エージェント s_j は、ユーザから与えられた締め切りまで、他の売り手エージェントの参加を待つ。

3 商品の交換に基づく売買交渉

第 2.3 節の (Step 4) において、もし、売り手エージェント s_j が、要求された個数以上の商品を持っていない場合、 s_j はその商品を持っている他の売り手エージェントを探し、商品の交換に基づく交渉を行う。2 つの売り手エージェントが交渉に成功した場合、それぞれのエージェントは、効用を増加させるこ

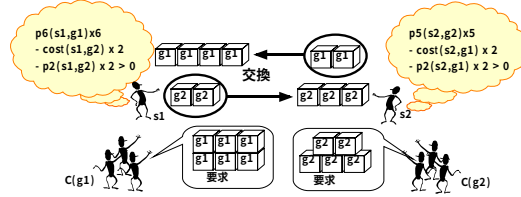


図 3 2 つの売り手エージェント間の交換機構

とができる。また、売り手間の交渉が成功することにより、多くの割引が期待できるため、買い手としても多くの利得を獲得することができる。以下に、2 つの売り手エージェント間での商品の交換に基づく交渉手法を示す。

図 3 に交渉の例を示す。本例では、2 つの売り手エージェント s_1 と s_2 が存在している。売り手エージェント s_1 は、商品 g_1 を 4 つ、商品 g_2 を 2 つ持っているとし、売り手エージェント s_2 は、商品 g_1 を 2 つ、および商品 g_2 を 3 つ持っているとする。ここで、買い手エージェントの提携 $C(g_1)$ が、売り手エージェント s_1 から商品 g_1 を 6 つ購入しようとしているとする。さらに、別の買い手エージェントの提携 $C(g_2)$ が、売り手エージェント s_2 から商品 g_2 を 5 つ購入しようとしているとする。

売り手エージェント s_1 は、十分な数の商品 g_1 を持っていないため、提携 $C(g_1)$ との売買を成立させることができない。そこで、売り手エージェント s_2 と交換に基づく交渉を行う。まず、売り手エージェント s_1 が次の提案を行う: 『 s_1 が商品 g_2 を 2 つ s_2 に渡す代わりに、 s_2 は s_1 に商品 g_1 を 2 つ渡す』。この時、以下の条件が成り立つならば、 s_1 と s_2 は提案に関して合意する:

s_1 について:

$$p_6(s_1, g_1) \times 6 - \text{cost}(s_1, g_2) \times 2 - p_2(s_1, g_2) \times 2 > 0$$

かつ、 s_2 について:

$$p_5(s_2, g_2) \times 5 - \text{cost}(s_2, g_1) \times 2 - p_2(s_2, g_1) \times 2 > 0$$

以上の条件の元でエージェントが合意することにより、交渉によって効用が減少しないことが保証される。商品を交換する際の条件の定義を以下に示す。

Definition 1 (商品交換の条件) 2つのエージェント s_1 および s_2 と, 2種類の商品 g_1 および g_2 が存在する場合に, 2つのエージェントは, 以下の 2通りのケースにおいて合意が可能である.

【case 1】

$shortage(s_1, g_1) = num(C(g_1)) - num(s_1, g_1) > 0$ かつ,

$shortage(s_2, g_2) = num(C(g_2)) - num(s_2, g_2) > 0$ として, s_1 の効用が,

$$u(s_1) = p_{num(C(g_1))}(s_1, g_1) \times num(C(g_1)) - cost(s_2, g_2) \times shortage(s_2, g_2) - p_{shortage(s_2, g_2)}(s_1, g_2) \times shortage(s_2, g_2) > 0$$

かつ, s_2 の効用が,

$$u(s_2) = p_{num(C(g_2))}(s_2, g_2) \times num(C(g_2)) - cost(s_1, g_1) \times shortage(s_1, g_1) - p_{shortage(s_1, g_1)}(s_2, g_1) \times shortage(s_1, g_1) > 0$$

である場合.

【case 2】

$shortage(s_1, g_2) = num(C(g_2)) - num(s_1, g_2) > 0$ かつ,

$shortage(s_2, g_1) = num(C(g_1)) - num(s_2, g_1) > 0$ として, s_1 の効用が,

$$u(s_1) = p_{num(C(g_2))}(s_1, g_2) \times num(C(g_2)) - cost(s_2, g_1) \times shortage(s_2, g_1) - p_{shortage(s_2, g_1)}(s_1, g_1) \times shortage(s_2, g_1) > 0$$

かつ, s_2 の効用が,

$$u(s_2) = p_{num(C(g_1))}(s_2, g_1) \times num(C(g_1)) - cost(s_2, g_2) \times shortage(s_1, g_2) - p_{shortage(s_1, g_2)}(s_2, g_2) \times shortage(s_1, g_2) > 0$$

である場合.

4 議論

4.1 パレート最適性

本論文で提案した交渉手法は, ゲーム理論 [17] における交渉問題のルール of 1つであると言える. ゲーム理論における交渉問題では, 交渉の結果得られる

配分を示す効用ベクトル $(u(s_1), u(s_2))$ が以下の (1), (2), および (3) の条件を満たす時, その配分はパレート最適であると言う.

1. $u(s_1) \geq u(s_1)^*$ かつ $u(s_2) \geq u(s_2)^*$. ただし, $u(s_1)^*$ および $u(s_2)^*$ は現在の状態を表す. すなわち, $u(s_1)$ と $u(s_2)$ は, 効用の最大値である.
2. $(u(s_1), u(s_2))$ は R の 1点である. R は, すべての可能な点の集合を表す.
3. R には, $u(s_1)' \geq u(s_1)$ かつ $u(s_2)' \geq u(s_2)$ となるような $(u(s_1)', u(s_2'))$ と異なる点 $(u(s_1)', u(s_2'))$ は存在しない. すなわち, $u(s_1)$ と $u(s_2)$ の両方がより大きくなる点は存在しない.

本論文で提案する交換に基づく交渉手法では, エージェントは 2つの戦略を持つ. 1つは交換すること, もう 1つは何もしないことである. ここで 2つの戦略を $str.1$, および $str.2$ とする. エージェントが交換によって合意に達した時, 得られる効用 $u(s_1)$ と $u(s_2)$ は, 明らかに最大化されるため, 条件 (1) が満たされる. また, 提案手法では, すべての可能な戦略集合は, $\{(str.1, str.1), (str.1, str.2), (str.2, str.1), (str.2, str.2)\}$ である. ここで, $(str.1, str.2)$ は, s_1 の戦略が $str.1$, および s_2 の戦略が $str.2$ であることを表す. 可能な効用ベクトルの集合は, $\{(u(s_1), u(s_2)), (0, 0), (0, 0), (0, 0)\}$ である. ここで, $(u(s_1), u(s_2))$ は, 戦略ベクトル $(str.1, str.1)$ を実行した場合の効用ベクトルである. $(u(s_1), u(s_2))$ は, R に含まれる点の 1つであるため, 条件 (2) が満たされる. 条件 (3) に関しては, 2つのエージェントは戦略ベクトル $(str.1, str.1)$ を選んだ時にのみ, 0 より大きい効用を得ることができる. どちらか 1つのエージェント, もしくは両方のエージェントが, 他の戦略を用いた場合, 効用ベクトルは $(0, 0)$ となり, 条件 (3) も満たされる. 以上より, 交換の結果得られる配分はパレート最適な配分になっていると言える.

4.2 実験

提案手法の有効性を示すために, 下記の設定で, シミュレーション実験を行った.

- 売り手の数: 5
- 買い手の数: 500

- 商品の値段: 200 to 300
- 商品の種類の数: 10
- 売り手の持つことのできる商品の最大数: 10
- 買い手の希望商品に関する分布: 正規分布
- 商品に対する売り手の効用の分布: 一様分布
- 売り手の商品数の分布: 一様分布

本実験では、既存の手法と提案手法とを比較した。既存の手法は、以下の 3 ステップからなる単純なプロトコルである。交換に基づく交渉手法は、第 2.3 節で示した (Step 1) から (Step 4) に基づくプロトコルである。

(Step A): 買い手エージェント b_k は、売買に関する要求 (商品 g_i , 希望価格 $r(b_k, g_i)$, および個数 $num(b_k, g_i)$) を宣言する。同じ種類の商品 g_i を希望するエージェントの集合が、提携 $C(g_i)$ を形成する。

(Step B): 売り手エージェント s_j は、商品 g_i に対する提携 $C(g_i)$ が存在し、売り手エージェント s_j の持つ商品の中に商品 g_i が含まれている、すなわち $g_i \in G(s_j)$ ならば、商品 g_i を売ることが可能である。もし、提携 $C(g_i)$ が存在しなければ、締め切り時間になるまで、自分が持っている商品の集合 $G(s_j)$ に含まれる商品に対する買い手エージェントの提携を探し続ける。

(Step C): $num(s_j, g_i) \geq num(C(g_i))$ が成り立つなら、売り手エージェント s_j は、商品 g_i を提携 $C(g_i)$ に売る。

本実験では、上で挙げたパラメータを変化させた、いくつかの異なる設定の元で比較を行った。実験では、各々の設定の元で 1000 回試行し、1 試行は 1000 単位時間とした。1 単位時間は、既存の手法では (Step A) から (Step C)、提案手法では、(Step 1) から (Step 4) とした。買い手エージェントと売り手エージェントは、 n 単位時間毎に電子マーケットに 1 つずつ参加するものとし、 $n = 1000 / (\text{エージェントの総数})$ とした。

図 4 は、売り手の効用の変化と売り手の数の変化の比較を示している。図 5 は、売れた商品の割合の

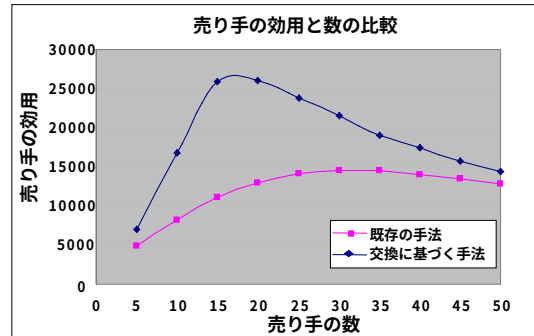


図 4 売り手の効用と売り手の数の比較

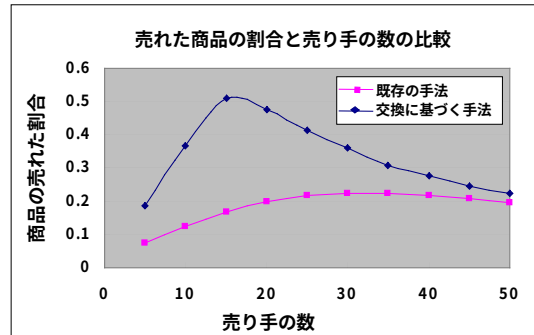


図 5 売れた商品の割合と売り手の数の比較

変化と売り手の数の変化の比較を示している。図 4、および図 5 が示す通り、提案手法の方が既存の手法と比較して、より多くの効用が得られている。特に、売り手の数が 10 から 25 の範囲では、既存の手法の 2 倍の効用が得られている。売れた商品の割合に関しては、提案手法では、既存の手法の 3 倍から 5 倍の効用が得られている。売り手の数が少ない時に、売り手の効用も売れた商品の割合も少ない理由は、売り手も買い手も数が少なすぎるために十分な数の商取引が行われないためと考えられる。

図 6 は、売り手の効用の変化と売り手が持つことのできる最大の商品数の変化の比較を示している。図 7 は、売れた商品の割合の変化と売り手が持つことのできる最大の商品数の変化の比較を示している。図 6 および図 7 から、提案手法では既存の手法より多くの効用が得られることが分かる。

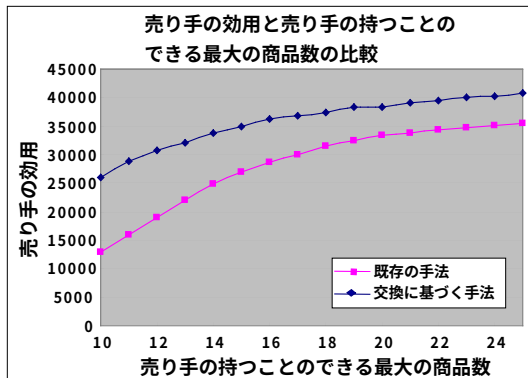


図 6 売り手の効用と売り手が持つことのできる最大の商品数の比較

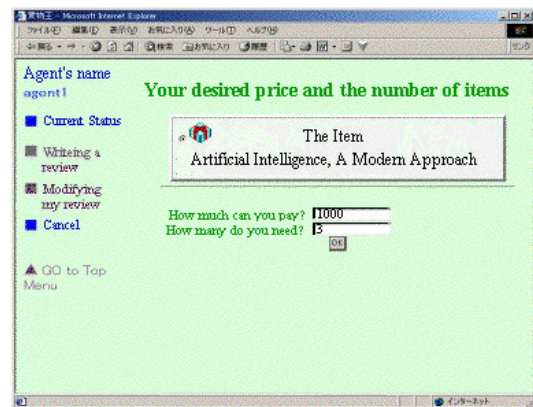


図 8 ユーザの希望価格と希望商品数の入力

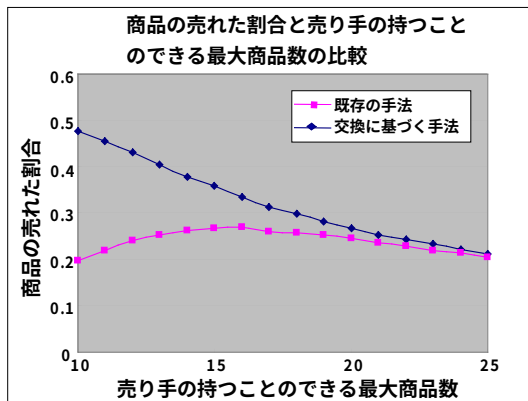


図 7 売れた商品の割合と売り手が持つことのできる最大商品数の比較

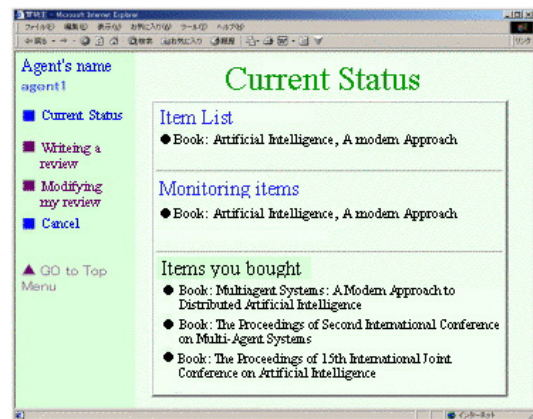


図 9 現在の取引の状況の表示

4.3 ユーザインターフェースの例

図 8 と図 9 は、G-Commerce のユーザインターフェースの例を示している。

図 8 のウィンドウは、ユーザの好み（嗜好）を入力するユーザインターフェースである。MiLog のエージェントは、CGI プログラムのように振る舞う機能を持つため、ユーザは、自分のエージェントに WWW ブラウザを用いてアクセスすることができる。ユーザは、自分の希望する商品、および希望価格を、図 8 に示すインターフェースの中央のテキストボックスから入力する。

図 9 のウィンドウは、現在の状態を示すユーザインターフェースである。ユーザは、現在、どの商品を監

視しているか、どの商品が売られているか、どの商品が売れたか、を知ることができる。また、商品を売りたいユーザもこのインターフェースを用いて商品の情報を入力する。買い手エージェントと売り手エージェントが合意した時、買い手エージェントは、ユーザに合意したことを知らせる。エージェントの合意後に、実際に購入を実行するのはユーザである。

5 関連研究

本節では、関連研究を示し、本研究との相違点を明らかにする。特に、インターネット上の複数のサイトから商品の情報を検索し統合するシステム、および、商取引を行う場を提供するシステムについて述べる。

インターネット上の複数のサイトから商品の情報を

検索し統合するシステムに関する研究または商用の製品としては Sherlock 2, AuctionWatch, BargainFinder, ShopBot, Jango などがある。Apple 社 [1] の Sherlock 2 は、インターネット上の複数のサーチエンジンにアクセスすることによって、複数の財に関する情報を得ることができる。AuctionWatch [2] は、主要なオークションサイトを継続的に監視し、ユーザから指定されたキーワードとマッチする商品が出品された時、ユーザに E-mail で知らせるというオンラインサーバである。BargainFinder [3] は、オンラインショップにおける商品の値段を比較し、買物における意思決定を支援するエージェントである。特定の商品を与えると、BargainFinder は前もって指定されたいくつかの商店の Web サイトから、その商品の値段を検索する。BargainFinder を発展させて開発されたショッピングエージェントが ShopBot [5] である。ShopBot の特長は、オンラインの商店のサイトの記述や、その商店に対するクエリを自動的に解析できる点である。ShopBot をさらに発展させた Jango [6] では、ユーザが商品に対する情報を指定することを可能にしている。BiddingBot [12] では、エージェントは複数のオークションサイトから情報を収集することだけでなく、実際に入札することも支援する。以上のシステムは、インターネット上の既存の WWW サイトから情報を収集することが主な目的であり、売り手と買い手双方の活動をエージェントが代行する本システムとは異なる。

オークション、または商取引を行う場を提供するシステムとしては、AuctionBot, eMediator, Kasbah, FishMarket, Tete-A-Tete, GroupBuyAuction などがある。AuctionBot [18] はオークションサーバである。AuctionBot のユーザは、商品を売るためにオークションを始めることができる。そのオークションでは、エージェントが売り手、もしくは買い手となり、あらかじめ定義されたプロトコルに従って入札を行う。AuctionBot の特長は、ユーザが自分自身のソフトウェアエージェントを作成できるような API を提供している点である。Kasbah [4] は、WWW 上の仮想的なマーケットプレイスであり、その上でユーザは、自分の代理として商品の売買を行うエージェントを生

成できる。Kasbah でのエージェント間の取引は単純なものである。また、AuctionBot のようにオークションのプロトコルに従う必要はない。FishMarket [15] は、仮想的なオークションの場を提供するシステムである。FishMarket では、ユーザはエージェントの入札戦略をエンコードすることができる。FishMarket は実世界のシステムとしては現在のところ使われていないが、様々な入札戦略を持ったエージェントによるトーナメントが開催されている。Tete-A-Tete [9] も仮想的なマーケットプレイスを提供する。Tete-A-Tete の特長は、エージェントが議論に基づく協調的な交渉を行う点である。eMediator [16] は、eAuctionHouse と eCommitter から成る電子商取引サーバである。eAuctionHouse では、Sandholm らによって開発されたアルゴリズムによって勝者を決定する、組合せオークションを行うことができる。eCommitter では、Leveled Commitment Contracts を使って最適な価格での契約を実現する。GroupBuyAuction [19] は、ボリュームディスカウントに基づく仮想マーケットである。ボリュームディスカウントとは、複数の買い手で多数の商品を共同で買う代償として、売り手が商品の単価を割引くことである。GroupBuyAuction では、買い手のエージェントの提携により、ボリュームディスカウントを実現している。以上のシステムは、仮想的な商取引の場を提供することにより、自動的な取引を実現しようという点では、本システムと同じである。しかし、本システムの相違点は、売り手が協調することができたり、P2P 技術で売り手と買い手のコネクションを確立する点である。

6 おわりに

本論文では、インターネット上の電子マーケット *G-Commerce* を提案し、システムにおけるエージェントの交渉方式について論じた。既存のエージェントに基づく電子マーケットシステムは、市場を管理するサーバを用いた中央集中型のシステムであった。従って、ユーザは、売買を行うためには、マーケットの場所を知り、その都度システムにアクセスする必要がある。さらに、サーバに対して売買に関する種々の情報を登録しておく必要があった。それに対

して、*G-Commerce* は特別なサーバを必要としない Peer-to-Peer(P2P) に基づく分散型のシステムである。*G-Commerce* のユーザは、個々にソフトウェアを起動し、インターネットに存在する多数のユーザの中から、適切な交渉相手から成る一時的な (ad-hoc) な電子マーケットを形成することができる。売買交渉は、エージェントを用いて行い、ユーザは交渉の過程で必要となった情報のみを表明するだけで良い。

また、既存の研究では、1 つの商品に関する、売り手と買い手の 1 対 1 の交渉プロトコルが提案されているが、インターネット上の多数のユーザを対象とする *G-Commerce* では、それら既存の交渉プロトコルでは不十分である。本論文では、複数の売り手と買い手がそれぞれに提携を形成することを前提とした交渉プロトコルを提案している。本論文では特に、売り手エージェントが提携を形成する手法として、商品の交換に基づく協調的な交渉手法について論じた。理論的／実験的な評価により、本手法によって、売り手と買い手の双方にとって有益な交渉が実現できることが示された。

参考文献

- [1] Apple URL. <http://www.apple.com/>.
- [2] AuctionWatch URL. <http://www.auctionwatch.com>
- [3] BargainFinder, <http://bf.cstar.ac.com/>.
- [4] Chavez, A. and Maes, P., "Kasbah: An Agent Marketplace for Buying and Selling Goods," In *Proc. of 1st International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents (PAAM-96)*, pp. 75-90, 1996.
- [5] Doorenbos, R. B., Etzioni, O. and Weld, D. S., "A Scalable Comprison-Shopping Agent for the World-Wide Web," In *Proc. of the 1st International Conference on Autonomous Agents (Agents-97)*, pp.39-48, 1997.
- [6] Etzioni, O., "Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web," In *AI magazine*, Vol. 18, No. 2, pp. 11-18, 1997.
- [7] Fukuta, N., Ito, T. and Shintani, T., "*MiLog*: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming," In *Proc. of the 1st Pacific Rim International Workshop on Intelligent Information Agents (PRIIA-00)*, pp. 113-123, 2000.
- [8] Gnutella URL. <http://www.gnutella.wego.com/>.
- [9] Guttman, R. H., "Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce," Master's thesis, MIT(Massachusetts Institute of Technology), 1998.
- [10] Guttman, R. H., Moukas, A. G. and Maes, P., "Agent-mediated Electronic Commerce: A Survey," *Journal of the Knowledge Engineering Review*, Vol. 13, No. 2, pp. 147-159, 1998.
- [11] 伊藤孝行, 新谷虎松, "協調的なスケジューリングのための交換条件の提示による説得手法," 電子情報通信学会論文誌 D-I, Vol. J81-D-I, No. 9, pp. 1099-1106, 1998.
- [12] Ito, T., Fukuta, N. and Shintani, T. and Sycara, K., "*BiddingBot*: A Multiagent Support System for Cooperative Bidding in Multiple Auctions," In *Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS-00)*, pp. 399-400, 2000(poster abstract).
- [13] Napster URL. <http://www.napster.com/>.
- [14] Oram, A., "Peer-to-Peer: Harnessing the Power of Disruptive Technologies," O'Reilly & Associates, 2001.
- [15] Rodriguez, J. A., Noriega, P., Sierra, C. and Padget, J., "FM96.5:A Java-based Electronic Auction House," In *Proc. of the 2nd International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents (PAAM-97)*, pp. 75-90, 1997.
- [16] Sandholm, T., "eMediator: A Next Generation Electronic Commerce Server," In *Proc. of AAAI-99*. AAAI Press, 1999. At the Intelligent Systems Demonstration Program.
- [17] 鈴木光男, "新ゲーム理論," 勁草書房, 1994.
- [18] Wurman, P. R., Wellman, M. P., and Walsh, W. E., "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents," In *Proc. of the 2nd International Conference on Autonomous Agents (Agents-98)*, pp. 301-308, 1998.
- [19] Yamamoto, J. and Sycara, K., "A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces," In *Proc. of the 5th International Conference on Autonomous Agents (Agents-01)*, pp. 576-583, 2001.

U-Mart: 経済学と工学をエージェントが結ぶ

喜多一 出口弘 寺野隆雄

Abstract: Agent-based simulation is one of promising approaches to understand and design complex socio-economical systems. U-Mart is a research program of providing economists and computer scientists with an agent-based simulation system of a financial market as a test bed for interdisciplinary study. This paper overviews aims, activities, achievements and future plan of the U-Mart program.

1 はじめに

インターネットに代表される情報通信革命のもとで社会経済システムのデザイン論として工学と経済学は互いに領域を重ねつつあり、システム工学、情報工学などの工学と経済学などの社会科学の学際的な交流が重要となっている。そこでは複雑な社会経済システムを研究対象とすることから、従来の数理モデルを用いた演繹的、理論的アプローチには限界があり、ブレークスルーのための新しいアプローチが求められる。

その一つの可能性がエージェントシミュレーションである。特に市場についてエージェントシミュレーションによる実験的研究は人工市場と呼ばれる [6]。こ

こでは取引エージェントの定型行動や学習、適応行動など従来の理論では扱えなかったエージェントの行動をシミュレーションし、それがいかなる価格形成をもたらすのかを実験的に考察する。また、人工市場や実市場に見られる複雑な現象への理論的接近として、統計物理学における相転移や臨界現象などの視点から研究を進める経済物理学 [7] とも交叉して興味深い研究が展開されている。

しかしながら人工市場の研究方法には実際の市場を模擬するのに十分な複雑さと計算機実験や分析が可能な単純さとを合わせ持つモデルの設計は必ずしも容易ではない。また理論モデルに比べ複雑にならざるを得ない実験設定やその結果について研究者間で効果的な知識の共有をはかる必要もある。このような課題に対して我々は工学と経済学が問題を共有するためのエージェントシミュレーションのテストベッド(試験台)を構築することで研究を促進することを考えている [3]。U-Mart と呼ぶ人工市場実験はその一つの試みとして進めているものである [8][9][10][12]。本論文では U-Mart 研究プログラムについて、その構想、実践と今後の展望について紹介する。

2 テストベッドとしての人工市場

2.1 研究プログラムの構想

U-Mart は市場について、その複雑性に起因するさまざまな問題 [1] を検討することを目指して構築されてきた。考察を行いたい問題としては例えば

1. 株式などの売買行動に関する人間集団の学習・創発や相互干渉、制度設計を含む意思決定問題の

U-Mart: Agent-based Simulation Interconnects Economics and Engineering.

Hajime KITA, 大学評価・学位授与機構, National Institution for Academic Degrees.

Hiroshi DEGUCHI, 東京工業大学, Tokyo Institute of Technology.

Takao TERANO, 筑波大学, The University of Tsukuba.

解明,

2. 市場における投機的売買行動と価格の乱高下などの市場の不安定性との関連の解明とこれを回避するための制度設計,
3. インサイダー情報の利用の検定など, 市況からのエージェントの行動の推定,
4. 取引戦略の市場における共進化的側面の実験, 等が挙げられる. これらの課題はいずれも従来の経済理論では対象とすることさえ困難なものであり, ある程度の複雑さを伴ったエージェントシミュレーションによる接近が必要である.

2.2 実験システムの要件

我々はテストベッドの構築にあたって以下の 3 点を要件とした.

実世界との関連付け : 経済システムについての実験として完全に架空の市場を設定することは, 現実経済のもつ基本的な複雑さや予期不可能性などの重要な要素が消去され, 実質的な意義のない仮想実験に終わりがねない. そこで, テストベッドは何らかの形で実世界と関連付けられていることが必要である.

広い参加可能性を持つシステムの開放性 : 経済学や金融工学, 計算機科学などの諸領域の学際研究を可能とするため, 多様な主体が参加可能な開放型システムとして設計する必要がある. これに関しては, Axelrod が行った繰り返し囚人のジレンマゲームについてのプログラムのコンテスト [4] やロボカップなどの事例が参考になる.

制度の解析と設計のための実験環境の提供 : 社会経済システムについて検討すべき課題は, 多数の自律エージェントの経済行動がシステム全体にどのような挙動をもたらすかを解析し, それらを律する市場の制度の影響を評価し, これを踏まえて制度を設計することである. テストベッドにはこのための実験的な考察を可能にすることが求められる.

3 人工市場システム U-Mart

3.1 U-Mart の特徴

先に挙げた要請に対して, U-Mart では以下のような方法を採用した.

まず, 実世界との関連付けとして, U-Mart では, 現実に存在する株価指数を対象とし, その「現物」ではなく, 現実には市場が存在しない「先物」を取引する市場を仮想的に形成しようとする^{†1} (図 1 参照).

先物市場では先物の売買契約は期日に現物市場の価格で清算される. したがって現物市場における価格の動き, それが将来, 現実の経済の動きにつれてどのように変化するかという期待についての判断と売買の決断への反映を迫られることになる. また, ソフトウェアエージェントの構成においては, エージェントに取引される商品の「価値」を組み込むことの難しさがある. 先物市場の導入により取引される商品の価値が期日において現物価格に拘束されることでこの難点を解消できる.

このように現実には存在しない仮想の先物市場を構成することにより, 仮想の市場での固有の取引, 価格形成が可能であり, その固有の挙動として現実の市場における判断の複雑さを十分に再現することが期待される.^{†2}

次に第二の要請については, U-Mart では, コンピュータ・プログラムによって自ら売買の判断をするソフトウェア・エージェントと自然人との両方の参加を可能とし, 同等に成果を競わせることを考えた.

人工市場の研究では売買戦略の (限定された) 合理性と多様性が重要である. U-Mart ではソフトウェア・エージェントのコンテストや人によるゲーミング・シミュレーションを通じて多様な戦略の可能性を探り, それらのもとでの価格形成の研究を可能にしようとする.

第三の要請については, 多様な取引戦略のもとで市場を仮想的に実験することにより, 実験結果の解剖学

^{†1} 先物市場の利用は U-Mart の提案者の一人である大阪市立大学, 塩沢由典教授の発案による.

^{†2} 証券会社などが提供しているいわゆる模擬市場でのゲームは価格形成機能を持たない.

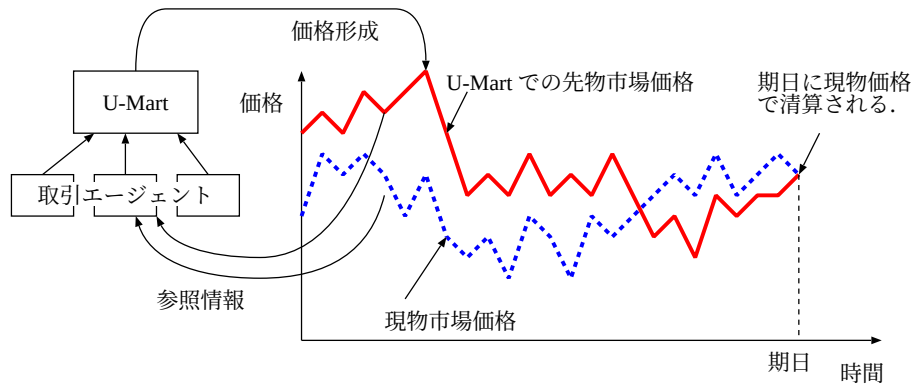


図 1 U-Mart における仮想先物取引。

的な分析が可能になる。そして値付け、決裁、手数量など市場のさまざまな制度の影響を評価し、制度の設計問題に接近しようとする。

4 U-Mart システムの概要

U-Mart では現実には存在する市場の「先物」を仮想の市場で取引する。システムはサーバ・クライアント型の形態を取り、TCP/IP 上に実装された専用プロトコルを用いて売買等の情報を交換する。取引所を模擬するサーバは各取引クライアントからの注文を受け付け、値付けと売買の約定を行い、資産口座を管理する。各クライアントはサーバから値動きなどの情報を得て独自の判断で売買注文を出す。また、U-Mart ではソフトウェアエージェントのほかに人間による取引も共存したゲーミング・シミュレーション型の実験も考慮して設計されている。U-Mart システムの詳細については文献 [9] を参照頂きたい。

4.1 U-Mart システムの構成

U-Mart の現在のシステムは以下のような文書とソフトウェアにより構成されている。

- Simple Virtual Market Protocol (SVMP) と実験用市場の仕様書。市場取引のための TCP/IP 上でのテキストベースのプロトコルを規定している。
- 仕様書に基づいて実装された 1 銘柄 1 限月の市場サーバ。Java を用いて記述されている。バックエンドのデータベースには PostgreSQL を採用

した。

- クライアント記述のための SVMP を呼び出す Java API。SVMP で規定されているリクエストとほぼ一対一に対応する API である。
- API を利用した U-Mart クライアントの実装例。ユーザが API を用いてクライアントの取引エージェントを作成するための雛型を提供している。
- サーバ管理のための GUI。Java/Swing で実装。
- ソフトウェア・エージェントのモニタリングと人による取引のための GUI。Java/Swing で実装されたものと C++ で Windows 上に実装されたものの 2 種類がある。
- クライアント記述のためのラッパープログラム
- クライアント開発のためのスタンドアロンサーバ API は SVMP のリクエストとほぼ一対一に対応しているため、初心者には戦略の記述が難しい。U-Mart への参加を容易にするため売買戦略を現物や先物の過去の価格系列や現在の資産状況から注文への関数と考え、これのみを記述すれば U-Mart クライアントが構成できるラッパープログラムとクライアント開発時にクライアントのテストを行うためにいくつかの簡単な取引エージェントを内部に組み込んだスタンドアロン型のサーバを開発した。

これらは SVMP の仕様策定に始まり、サーバの実装から徐々に発展してきたものである。その過程で 2 度の公開実験や大学・大学院での教育などの実践を経て必要なソフトウェアが認識、作成されるとともに改

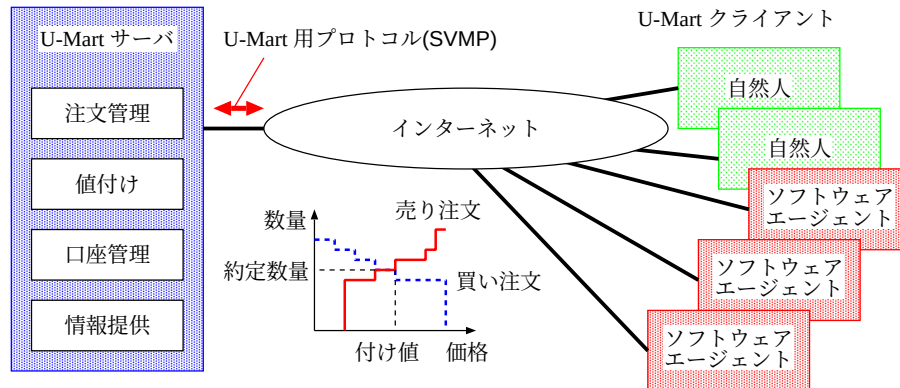


図 2 U-Mart の概要.

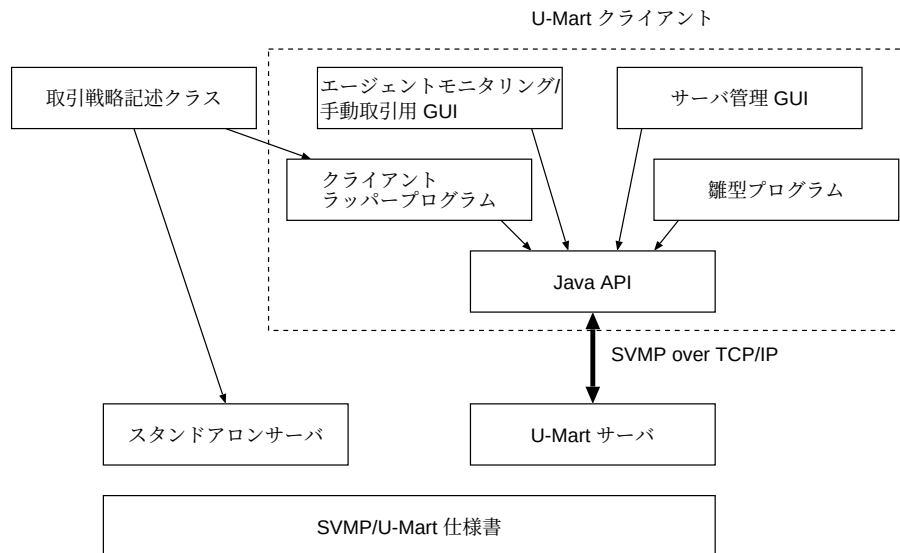


図 3 U-Mart システムの構成.

良が進められてきた.

4.2 次世代 U-Mart システムの構想

現行の U-Mart システムは 1 銘柄 1 限月の市場を前提とし、また U-Mart の実験を経験する前に概略の設計がなされた。その後の実践を経て SVMP の拡充、バグの修正や実行効率の改善などがなされたが、実験の仕様の拡大などの要求にはシステムの再設計が必要であり、次世代の U-Mart システムについて以下の点を課題に作業を開始した。

- 複数銘柄・複数限月の市場を構成可能とすること。これにより、より多様な取引戦略が取り込めるとともに、結果のより本格的な分析が可能になる。
- サーバ負荷軽減のための SVMP リクエストの見直し。
- ロギングのためのデータベース運用の柔軟化。
- 市場の取扱効率の向上。

現在、SVMP の見直しを済ませ、サーバについてのオブジェクト設計も概ね完了した。今後、サーバと

クライアント API の実装, GUI やクライアントの移植などを進めて行くとともに, 複数銘柄, 複数限月の市場での実験計画を検討して行く予定である.

5 U-Mart 公開実験

U-Mart ではすでに 2 度にわたって取引エージェントを公募して公開の実験を行った.

まず, 2000 年 8 月にこのような実験のフィージビリティを探ることを目的に Pre U-Mart 2000 を, 2001 年 8 月により本格的な公開実験として U-Mart 2001 をいずれも計測自動制御学会主催の合宿型研究集会「創発システムシンポジウム」の場を借りて行った.

Pre U-Mart 2000 の模様については文献 [10] に報告しているが, 主催者の予想を上回る多様な取引戦略や開発方法があることが認識された. 実際に行った実験では市場が初期段階で猛烈な乱高下を示し, その原因追求などを通して市場やそこでの取引戦略についての理解が深まった. また経済学系, 工学系両方からの参加者が合宿という場を共有して共通のプログラムに取り組むことが相互理解に極めて有効であることも改めて実感された.

U-Mart 2001 では Pre U-Mart 2001 の経験を踏まえ, U-Mart を用いたより本格的な研究を支援するためにエージェントのプログラムと説明文書の事前提出を求め, シンポジウムに先だって現物価格や乱数の系列を変えた実験を行いその結果を取りまとめて報告するとともに, 事前実験で優秀成績を示したエージェントに絞り, 人間による取引も加えて当日, 会場でデモンストレーションを行った. U-Mart 2001 については今後, 参加エージェントのすべてのプログラムや実験結果などを整理して提供して行く予定である.

6 U-Mart の教育面での利用

研究プログラムとして進めて行く過程で, U-Mart システムは経済学, 工学の両分野での教育に極めて効果的な教材となることが理解されてきた. そして実際に U-Mart を用いた教育がいくつかのサイトで実践

されている^{†3}今後, 教材としても利用しやすい形でドキュメントやソフトウェアの積極的な提供を進めて行く予定である.

6.1 経済学系分野での利用

経済学系 (経済学, 経営学, 商学などの学部, 大学院) の分野では U-Mart は

- 市場が個々の取引エージェントの行動により形成されるという基本的な概念を実験的に示すことができる,
- 金融市場, とくに金融派生商品 (デリバティブ) 市場の考え方や制度を体験的に理解することができる,
- 個々のエージェントの戦略や行動が市場に与える影響を体験することができる,

などの点で有効な教材である.

利用方法としてはもっぱら学生自身が (公開実験などで提供されたエージェントとともに) 市場取引に参加するゲーミング・シミュレーションが用いられる. 京都大学の事例は文献 [10] に紹介されている. また, ある程度, 数式や計算機に馴染みさえあれば自ら U-Mart システムを用いてソフトウェア・エージェントを構成することも社会科学系の学生でも十分に可能である.

6.2 工学系分野での利用

一方, U-Mart は情報工学系の分野での教材としても

- 学部レベルの初級プログラミングの課題,
 - 大学院レベルでの予測, 学習, 適応, 最適化アルゴリズムやエージェントシミュレーションそのものの学習,
 - 工学系の学生に経済学や金融の紹介,
- などに有効である.

とくにプログラミングの課題としては,

^{†3} U-Mart による授業が実施されたサイトとしては経済学系では京都大学経済学部/大学院経済学研究科, 大阪市立大学経済学部, 大阪産業大学経済学部, 筑波大学大学院経営政策科学研究科, 中央大学商学部など, 工学系では徳島大学工学部, 東京工業大学大学院総合理工学研究科, 東京大学工学部などである.

- 自らの知識やスキルのレベルに応じた取り組み,
- 自らの着想をプログラムコードとしての実現,
- 事前の予想のもとに作成したプログラムの実際の複雑な環境での動作の確認,

といった点で主体的に学習できることが通常のプログラミングの演習とは異なる点であり, 機械系の学科などで見られるロボットコンテストと同様の効果を持っている. また U-Mart のエージェント開発は個人の単位で比較的短期間に組み組めるためカリキュラム編成上も扱いやすい.

7 おわりに

U-Mart は情報通信革命のもとでの社会経済システムのデザインという研究対象の重なりを意識し, 経済学と工学の共同研究の場としてエージェントシミュレーションのテストベッドを提供する試みとして始められた研究プログラムである. U-Mart は当初の予想以上に多くの方々に関心を持って頂いたり, 教育面で極めて効果的な教材となり得ることの発見など当初の予想を上回る成果を挙げている.

一方で, 実際に安定に稼働するシステムを構築し, 多くの方々の協力を得て多様な戦略を持つ取引エージェントを構成することなどはようやく本格的な研究レベルの分析に着手できるところまでたどりついたという感がある. 今後, 研究のプログラムをより明確, 具体的なものとして進めて行く予定である.

U-Mart 自身は金融市場を例にテストベッドを提供しようという試みであるが, その経験からもエージェントシミュレーションは社会科学の広範な領域で新しい研究アプローチとなり得ると期待できる. しかし, それは, モデリング, シミュレーション, 分析などす

べての面において従来の社会科学の理論研究や実証研究とも, また自然科学の応用という考え方に重心を置いて来た工学の研究とも異なった発想や行動規範が求められる. このような点を明確にしつつ有効な学際研究を進めて行くことがこの方向の研究を成功に導く鍵となるであろう.

最後に大阪市立大学 塩沢由典 教授をはじめ U-Mart 研究会に参加されている皆様に感謝の意を表します.

参考文献

- [1] 塩沢由典:「複雑系経済学入門」, 生産性出版 (1997)
- [2] 高木, 木嶋, 出口他:「マルチメディア時代の人間と社会」, 日科技連出版 (1995)
- [3] 出口:「複雑系としての経済学」, 日科技連出版 (2000)
- [4] Robert Axelrod: The Complexity of Cooperation — Agent-Based Models of Competition and Collaboration, Princeton University Press (1997)
- [5] 塩沢由典:進化経済学の可能性としてのバーチャル市場の参加型シミュレーション提案, 進化経済学会第 3 回大会予稿集, (1999.3)
- [6] 和泉, 植田:人工市場入門, 人工知能学会誌 15-6, pp. 941-950 (2000).
- [7] 佐藤, 高安:統計物理から見た人工市場,, 人工知能学会誌 15-6, pp. 958-965 (2000).
- [8] 塩沢, 出口, 喜多, 寺野, V-Mart 研究の目指すもの, SICE システム情報部門合同シンポジウム 99, 169/174 (1999)
- [9] 佐藤, 福本, 廣岡, 久保, 生天目, 人工市場のシステム構造, 人工知能学会誌 15-6, pp. 974-981 (2000).
- [10] 佐藤 浩, 松井 啓之, 小野 功, 喜多 一, 寺野 隆雄:オープン型人工市場におけるエージェントの戦略と市場の挙動 (Pre U-Mart 2000 実施報告) SICE 第 13 回自律分散システム・シンポジウム, pp. 203-208 (2001).
- [11] <http://www.u-mart.econ.kyoto-u.ac.jp/> キャピタル (1998).
- [12] Hiroshi Sato, Hiroyuki Matsui, Isao Ono, Hajime Kita, Takao Terano: U-Mart Project: Learning Economic Principles from the Bottom by both Human and Software Agents, Proc. of the 1st Int'l workshop on Agent-based Approaches in Economic and Social Complex Systems, Matsue, pp. 59-66 (1991).

エージェントシミュレーションによる 「チープ・トーク」モデルの拡張

小山 友介

1 やや長い前書き：社会科学のエージェントシミュレーション

社会科学のエージェントシミュレーションは、個々のエージェントが環境を認識して意思決定をする多主体複雑系特有の相互作用の難しさ (出口 [4]) をどうモデル上で実現し、分析するかが課題となる。モデル上で実現する方法および分析対象は広範囲にわたり、明確な基準はない。単純な指標として、モデル内の「エージェント数」とモデルと個々のエージェントの行動原理の「複雑性」(これも定義が厄介な基準である) の積を考えてみるとわかりやすい。指標の大きさが水準以上に達する現象を分析の対象にしていると言っていいだろう。もし指標の大きさが明確に定義できれば、指標の大きさそのものが「研究フロンティア」の位置を示していると考えてよい。

この視点からは、社会科学のエージェントシミュレーションの研究は「エージェント数」と「複雑性」を両極にした研究フロンティア上に位置する事になる。可能な限り行動原理をシンプルにしてエージェント数を増やし、多数のエージェントによる相互作用から発生した創発現象を研究する経済物理学が一方の極

である。もう一方の極は、コンピュータを用いたエージェントシミュレーションではないが、実際の人間数人が複雑な現象について行うゲーミングシミュレーションを挙げる事ができるだろう。

これまで社会科学のエージェントシミュレーションが採用してきた研究パラダイムは、Axelrod [3] が主張した KISS 原理 (Keep it simple and stupid) である。経済物理学で扱うような数万エージェントは扱わないが、数十～数百のエージェントを扱うために、行動原理はできるだけシンプルにし、相互作用のあり方もできるだけ単純なものに限定する。エージェントが相互作用する空間は用意しないでランダムマッチングにするか (Axelrod [3] や多数の生態学からの派生モデル)、セル型の空間内での近隣間相互作用に限定する (Epstein [5]、Gaylord et.al [7] など) か、市場タイプの相互作用のみに集中する (Arthur [1] など) と言った形でモデル自体の複雑性をできるだけ減らすようにしていた。こういった事情で、社会科学のエージェントシミュレーションのほとんどは、Axelrod [3] に始まる GA (遺伝アルゴリズム) 系とシェリング [9] に始まるセル系、Arthur [1] に始まる人工市場 (ここに経済物理学も含まれる) に分類する事ができる。

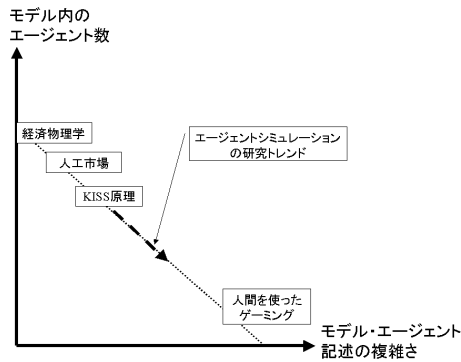
KISS 原理に基づく多数のエージェントによるシンプルな相互作用モデルの研究は、研究から得られる成果が減少しつつあるため、今後はエージェント数が少なくてもより複雑な相互作用を行うモデルへと研究トレンドがシフトしてゆく事が予測される (図 1)。このような複雑で具体的なモデルを構想するにあたっては、実際の現象や実験データ、ゲーミングから得ら

The extention of the 'cheap-talk' model in the agent simulation

Yusuke Koyama, 日本学術振興会特別研究員・京都産業大学客員研究員, Research Fellow of the Japan Society for the Promotion of Science, Visiting Fellow of Kyoto Sangyo University

koyama@cc.kyoto-su.ac.jp / yus@ma1.seikyoku.ne.jp ,
<http://ha1.seikyoku.ne.jp/home/yus/index.html>

図 1 研究トレンド



れた知見などからモデルの構想力を豊富にする事が必要である。最終的に一番重要なのは、研究の蓄積やモデリング手法のブレークスルーによって、研究可能フロンティアを図 1 の右上方向に広げる事である。そのためには、技術に比較優位を持つ工学系と、ビジョンや知識に比較優位をもつ社会科学系の共同研究が今後ますます重要になってくるだろう。

2 協力発生の難しさ

2.1 事前コミュニケーションの重要性

ゲーム理論の囚人のジレンマ (表 1) の例で有名なように、見知らぬ他人と協力する事は難しい。自分の利益を最大化する個人を仮定すると、ジレンマ状態では協力は発生できない (ワンショット・ゲームでのナッシュ均衡)。同じゲームが繰り返され、将来利益の割引評価率が十分大きければ、協力状態もありうるが、協力の発生には (比喩ではなく、本当に) 無限回の繰り返しが必要である (フォーク定理)^{†1}。

表 1 囚人のジレンマ

		プレイヤー B	
		協力	裏切り
プレイヤー A	協力	(3,3)	(0,5)
	裏切り	(5,0)	(1,1)

†1 繰り返し囚人のジレンマと協力の発生については、Axelrod [2] が今でも基本文献である。社会心理学者による議論は山岸 [11]、社会学者による議論は盛山他 [13]、経済学者による議論は松島 [8] を参照。

理論的なフレームワークを離れて、人間による囚人のジレンマ実験の結果 (Frank [6]) をみてみよう。質問紙調査の結果によると、過半数の被験者が「相手が協力的に行動する確信があれば、自分も協力的に行動する」と答えている。実際の実験結果では、相手との選択前コミュニケーションを被験者に許可した場合、許可しない場合に比べて協力を選択する確率は大きく上昇する。さらに、「お互いに裏切らない」という (拘束力のない) 約束を許可されている場合、約束ができない場合より協力の選択率は高くなる。この結果は、実際の行動前にコミュニケーションがとれ、相手が協力する確信を得られれば、協力が起こりやすくなる事を示している。

2.2 「チープ・トーク」による相手の情報の収集

実際の現実世界では、ジレンマ状態の相手と約束が可能とは限らない。コミュニケーションできない場合も多い。しかし、われわれは直接コミュニケーションが不可能でも、何らかの形で相手の情報を収集し、また、無意識のうちに自分の情報を発信している^{†2}。このことがコミュニケーションに準ずる効果を挙げている。

「ゼロコストもしくは無視可能な低コストでの相互情報交換」を、ゲーム理論ではチープ・トーク (Cheap Talk) と呼ぶ。チープ・トークで得る情報は、直接コミュニケーションが難しい実際の社会で協力が発生するために、非常に重要な役割を果たしている。

2.2.1 チープトークによる社会秩序の変化の例：Secret Handshakes

ここで、チープ・トークを用いた Secret Handshakes という簡単なモデルを考えてみよう^{†3}。多数のプレイヤーの一群の中からランダムに 2 名がマッチングされ、ゲームを 1 回プレイする。利得行列は表 2 を用いる。

利得行列 (表 2) にはナッシュ均衡が 2 つある。

†2 話し方や服装、しぐさなどの「自分がその場にいるだけで近くの人に伝達される情報」の量は膨大である。無意識に相手から収集する情報、こちらが発信する情報は意外に多い。

†3 詳細は Weibull [10] 参照。

表 2 コーディネーション・ゲーム

		プレーヤー B	
		選択肢 A	選択肢 B
プレーヤー A	選択肢 A	(3,3)	(0,0)
	選択肢 B	(0,0)	(1,1)

{ 選択肢 A, 選択肢 A } と { 選択肢 B, 選択肢 B } である。それぞれを均衡 A と均衡 B と呼ぶとき、均衡 A が均衡 B をパレート優越 (全員の利益が上回る状態) している。

プレーヤー群全体が選択肢 B を採用している状態、すなわち社会状態が均衡 B の場合、たとえ均衡 A の方が望ましい事がわかっていても、(利得を得るためには) プレーヤーは選択肢 B を選ばざるを得ない。すなわち、社会状態が均衡 B の時には選択肢 A は進入不可能^{†4}である。

このとき、プレーヤー群の一部が突然変異を起こしたとしよう。一部のプレーヤーは行動を選択する前に、一部の突然変異を起こしたプレーヤーのみにわかるメッセージ μ を出す。変異プレーヤーは、相手からメッセージを受け取った時のみ (すなわち、相手も自分と同種である事を確認できた時のみ) 選択肢 A を選択し、それ以外のときは選択肢 B を選択する。事前にメッセージを交換するこの戦略は均衡 B の中に進入可能であり、徐々に環境内で勢力を増し、環境全体を均衡 A へと移行させることが可能である。チープ・トークによって社会状態が良くなる方向へと移行させる事が可能になるのである。

しかし、Secret Handshakes モデルは囚人のジレンマではなく、コーディネーション・ゲームである。行動主体に利益最大化行動を仮定している限りでは、チープ・トークが可能であっても、囚人のジレンマ (およびその多人数版の社会的ジレンマ) を解消する事は原理的に不可能である。Secret Handshakes モデルでは複数のナッシュ均衡状態から望ましい均衡へと移行させる事はできても、ナッシュ均衡戦略以外の行動をとらせることはできない。

^{†4} 選択肢 A も選択肢 B も進化安定戦略 (Evolutionary Stable Strategy: ESS) である。進化安定戦略についての詳細は、前述の Weibull [10] を参照。

ジレンマ状態で協力を発生させるには、先ほどの実験の結果で見受けられるような何らかの先験的な協力性向を想定し、短期的な利益最大化 (いわゆる合理性) の仮定を外す必要がある。節を変えて、より現実的な方向への議論の拡張を試みてみよう。

2.2.2 「社会的知性」

チープ・トークによって Secret Handshakes モデルのような明瞭なメッセージを交換することは、現実では難しい。通常は、相手から受け取った不明瞭なメッセージを解釈する必要がある。いわゆる「鈍感な人」が多数いる事からわかるように、メッセージの解釈能力には個人差がある。この「メッセージの誤解」も考慮に加えると、チープ・トークはノイズが非常に多く、使うのが大変なコミュニケーションである。

さらに、相手を利用して自分が得をしようと考えている人も多数存在する。いわゆる詐欺のように、コミュニケーションを逆に利用して相手に信用させ協力させ、裏切ると言う可能性もある。チープトークで得た「情報」を受信した側で適切に解釈できるかどうかは、協力が発生できるか否かだけではなく、迂闊に悪人を信頼してしまって自分が「食い物」にされるか否かと言った、より根源的な場面で重要である。

この「誰が信頼できるか」を見抜く能力を、山岸 [12] では「社会的知性 (social intelligence)」と呼んでいる。「社会的知性」が高い人ほど、裏切られて利用されかねない環境で協力的な人とそうでない人を見抜き、協力的な人を信頼する能力が高い。実験の結果からも、われわれの日常経験からも、「相手が協力的なら協力的にふるまう」人が社会に (それなりに) 多数存在する事はわかっている。こういった協力的で社会的知性が高い人同士がチープトークで情報交換しあうことで、社会内では協力が発生しているという「お話」を、われわれは想像する事ができる。

2.3 本研究の目的

山岸 [12] によると、他者を信頼する傾向が強い人は相手が信頼できる人 (= 協力してくれる人) を見抜く能力が強く、社会内に協力できる人を見つけ、ネットワークを広げてゆける人である。一方、社会的知性が低い人は、信頼できる人を見つけ出す能力が低い

で、すでに存在している協力のネットワークにしがみつく (コミットメント状態) 以外できない人である。

山岸 [12] の知見は質問紙調査と実験から得られたものである。この現象をエージェントシミュレーションで再現を試み、エージェント間の相互作用によって他の収束がありうるかを探る事が本研究の目的である。

3 モデル

3.1 内容

n 人のプレーヤー^{†5}によるグループを想定する。各プレーヤーに社会的知性、倫理度 (自分から裏切らない性向。Axelrod [2] の意味での「上品さ」)、基本信頼度 (初対面の相手に対する信頼度)、信頼の修正速度をエージェント固有パラメータとして用意する。毎回ランダムに 2 人をマッチングさせ、ゲーム (後述) をプレイする。ゲーム後相手への信頼度を修正し、双方協力の場合には他プレーヤーの情報を交換し合う。このプレイを繰り返す。

ゲームは 2 プレーヤー・ 3×3 の対称ゲームである。利得行列は表 3 である。表 3 の利得行列の純粋戦略ナッシュ均衡は M,M であり、ESS でもある。各プレーヤーは積極的に相手と協調し高い利益を得ることを目指す協調行動 C、C を選択する相手を利用して高い利益を得ようとする D、相手と積極的に関係を持たない選択肢 M という 3 つの選択肢を持っている。

表 3 チーブ・トーク・ゲーム

		Player 2		
		C	M	D
Player 1	C	(30,30)	(10,13)	(0,50)
	M	(13,10)	(16,16)	(13,10)
	D	(50,0)	(10,13)	(5,5)

プレーヤー i は以下のパラメータを保有する。 I_i, E_i, R_{ij}, V_i は 0 から 100 の整数である。

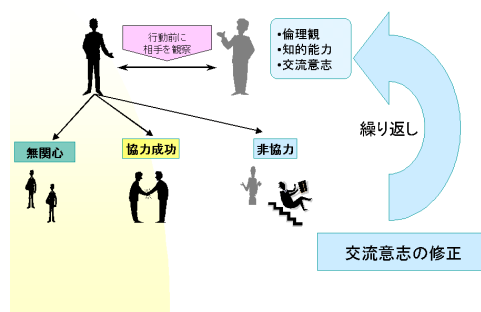
- 知性 I_i : 大きいほどゲームをプレイするときに

相手の協力度 (後述) を正確に推測できる。

- 倫理 E_i : 大きいほど各プレイで協力行動を採用する確率が大きくなる。
- 信頼度 R_{ij} : 相手が協力行動を取る信頼度。初対面時はデフォルト (R_i) で、毎回のプレイ毎に修正する。
- 信頼度修正速度 V_i : 大きいほど相手のプレイ結果に従って信頼度 R_{ij} が大きく変化する。
- 履歴 : 各相手のプレイ回数、相手が C を出した回数、D を出した回数を記録する。

一回のマッチングで、以下の一連の行動が行われる。これを繰り返す (図 2)。

図 2 チーブトークゲーム



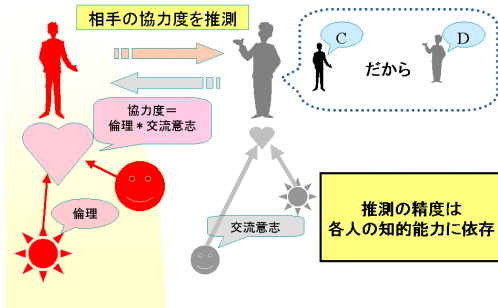
1. プレイする 2 プレーヤーを n 人からランダムに選ぶ。
2. 行動前に相互に観察し、相手の協力性向 (後述) を推測する。推測の精度は自分の知性 I_i に依存する。
3. 読み取った結果を元に自分の行動 (C,M,D) を決定する。行動の結果、得られる利得が双方に与えられる。
4. プレイ後、相手への信頼度を修正する。修正度合いは信頼度修正速度 V_i に依存する。
5. 二人の行動がともに協力 (C,C) の場合、評判情報 (後述) を交換し、信頼度を修正する。

^{†5} ゲーム理論の慣例に従い、エージェントではなく「プレーヤー」と書くことにする。意味に違いはない。

3.2 行動決定およびパラメータ修正方法

ゲーム中の行動決定では、(モデル内で定義する) 協力性向の大小で決定する。協力性向が大きい時は、相手が裏切らない限り協力を試みる。小さい時は、相手に対し攻撃的でスキがあれば利用しようとし、相手の利得を下げるには若干のコストも平気^{†6}である。

図 3 行動決定



選ばれた 2 プレーヤーを $i, j (i > j)$ とする。1 から 100 までの整数値をとる乱数を $RAND100$ とする。プレーヤー $\alpha (= i, j)$ が相手の意図を読み取ろうとする行為を S_α であらわす。

1. 自分の倫理 E_i から $RAND100$ を引き、相手への信頼度 R_{ij} を掛け合わせたものを自分の協力性向 X_i とする。

$$\begin{cases} X_i = (E_i - RAND100) \cdot R_{ij} \\ X_j = (E_j - RAND100) \cdot R_{ji} \end{cases} \quad (1)$$

2. 相手の X を推測したものを Y とする。

$$\begin{cases} Y_i = S_i[X_j] \\ Y_j = S_j[X_i] \end{cases} \quad (2)$$

Y の推測は 2 段階で行う。

- (a) X_j の符号を推測する。最大推測誤差は $50 + I_i/2\%$ 。すなわち、知性が 100 の時は完全に把握できる。
- (b) 相手の X_j の実値を $\pm(100 - I_i)\%$ の精度で推測する。 $I = 70$ のときは推測値は X_j

の 70% から 130% である。

3. 相手の Y_j を推測したものを Z_i とする。推測方法は 2 と同様とする。

$$\begin{cases} Z_i = S_i[Y_j] \\ Z_j = S_j[Y_i] \end{cases} \quad (3)$$

4. Y_i と Z_i を表 4 と照らし合わせ、相手の行動 A_j を推測する。

表 4 相手の行動の推測

		Z_i		
		300～	-300～300	～-300
Y_i	300～	C	M	M
	-300～300	C	M	M
	-300	D	D	M

5. X_i および A_i を表 5 と照らし合わせ、自分の行動を決定する。

表 5 自分の行動の決定

		A_i		
		C	M	D
X_i	300	C	M	M
	-300 300	C	M	M
	-300	D	D	M

6. プレイ後に今期 t の現在の相手への信頼度を $R_{ij,t}$ を修正する。

$$R_{ij,t+1} = \left(\frac{1 - V_i}{100} \right) \cdot R_{ij,t} + kV_i \quad (4)$$

k は相手が C を選択時は 1、M では 0.6、D では 0 とする。

7. 双方の行動が (C,C) の場合、情報を交換し、信頼度を以下の手順で修正する。

- (a) 双方が過去のプレイの履歴から、C を採った回数が D を採った回数より多かった相手プレーヤーのリストを作成する。
- (b) リストを相互に交換し、そのリスト内の相手への信頼度を実際にプレイした場合と同様に (下方に) 修正する。

^{†6} 実験では、相手より自分の利得を大きくしたい、という行動 (Spite 行動) を取る人が存在する。この仮定はそれに倣ったものである。

5 実験結果 (2) 知性混合グループ・修正速度も学習させた時

グループ内に知性の異なるメンバーが存在し、修正速度も学習させた場合を考えてみよう。 $I = 90$ (高知性) と $I = 30$ (低知性) のプレイヤーを用意し、その他のパラメータの初期値は $E_i = 60, R_i = 80, V_i = 15$ で共通にした。ゲーム 45 回 を 1 セットとし、セット終了時に 1 マッチング辺り利得が最下位のプレイヤー a と最高のプレイヤー b の E_b, R_b, V_b を変化させたミュータントとを入れ替える作業を 10000 回繰り返した (ゲーム 45 万回が 1 実験である)。調べた条件は、表 6 の通りである。

表 6 実験のうわけ

	低知性	高知性
実験 2-1	6 人	
実験 2-2	4 人	2 人
実験 2-3	2 人	4 人
実験 2-4		6 人

5.1 発生した戦略の分類

4 通りの実験をから発生した戦略を全て取り出して分類すると、表 7 のようになった。

表 7 発生した戦略

	倫理	初期信頼	修正速度
協力 & 警戒 (WR)	高	高	高
2 度目に期待 (ES)	高	低	高
蜜月 (HM)	高	高	低
Hit & Run(HR)	低	高	高
交渉拒否 (RC)	任意	低	低

それぞれの戦略の具体的な特徴は、以下のようになる：

- 「二度目に期待」型 (ES)：高倫理、低初期信頼、中～高修正速度。自分からは裏切らないで一度様子を眺め、相手が裏切らなかった場合には、一

気に友好的になる。

- 「蜜月」型 (HM)：高倫理、高初期信頼、低修正速度。最初から積極的に協力し合い、少々の失敗では相手への信頼が揺るがない。
- 「協力 & 警戒」型 (WR)：高倫理、高初期信頼、高修正速度。最初から積極的に協力を求めるが、相手の出方次第でさっと引いてしまう。Axelrod [2] の「しっぺ返し (Tit For Tat)」に近い戦略。
- 「Hit and Run」型 (HR)：中～低倫理、高初期信頼、高修正速度。交流を行う相手には裏切り、相手が攻撃的になったらすぐに交流を止める。
- 「交渉拒否」型 (RC)：低初期信頼、低修正速度。初期信頼を低い水準でずっと保つ戦略。倫理は中 (50 前後) から低い事が多いが、基本的にはどんな値でも構わない。

基本的に、上の 3 つが協力が成功した場合、下の 2 つが協力が失敗した場合と言っていいだろう。戦略としてかなり優秀なのは ES と WR で、この 2 戦略は協力を形成する事が可能であるのに加えて、相手にだまされつづける事がない。双方の違いは初対面の相手への積極性にある。HM は HM ばかりが仲間にいる場合にはノイズに対する耐性が強いので協力形成に有利だが、相手が HR の場合には延々と搾り取られる可能性がある。

やや例外なのは RC で、「成功もしないが失敗もしない」戦略である。相手を信頼しない状態を保つ戦略で、行動決定には倫理はほとんど関係ないが、戦略が RC に固まるまでには多数裏切り・裏切られた経験をもつ (=協力を出さないように相互に学習しあう) ので、周囲の倫理は低くなっている場合が多い。

5.2 それぞれの実験での結果

次に、それぞれの実験の結果を見てみよう (表 8)。

グループが全て同じ知性で占められている場合 (実験 2-1・2-4) では過半数が協力が成功 (=戦略が HM, ES, WR) している。その点では、同質的なグループで協力が発生する／しないと言う視点からは知性は大きな影響を与えていない事がわかる。

しかし、発生した戦略をくわしく眺めると、低知性

表 8 実験結果 (2)

	2-1	2-2		2-3		2-4
	低	低	高	低	高	高
1	RC	WR	RC	ES	HR	HR
2	HM	WR	WR	ES	RC	HR
3	ES	HR	HR	RC	ES	WR
4	ES	HR	HM	ES	RC	WR
5	RC	HR	WR	HR	WR	HR
6	ES	ES	HM	ES	ES	HM
7	ES	ES	HR	WR	ES	HM
8	ES	HR	HR	RC	RC	HR
9	HR	HR	HR	RC	ES	WR
10	ES	HR	HM	HM	WR	HM

の場合 (実験 2-1) では ES がほとんどであり、協力の形成に慎重である。また、一方、高知性の場合 (実験 2-4) では WR^{†9} と HM が 3 回づつであり、積極的に協力行動を採る事が見受けられる。

グループが低知性と高知性に分かれている場合 (実験 2-2・2-3) では、協力発生が非常に難しくなっている。加えて、協力失敗の現れ方が異なっている。

低知性者が多数派の実験 2-2 では、双方が協力的な戦略となった事例が少ないだけでなく、HR が多数発生しており、協力的な人間を騙して利益を得ようとする事例が多数発生している事がわかる。高知性者が多数派の実験 2-3 では、HR ではなく RC が多数発生しており、コミュニケーションを閉ざす事で「万人の万人に対する闘争状態 (ホブズ)」を回避する方向に進化している。また、実験 2-2・2-3 の双方ともに、協力的な戦略を採用しているのは高知性者が多い。

6 結論と課題

今回の実験では、知性の高低の区別なくエージェントは協力を試みるが、協力へのアプローチの仕方に違いがあった。社会的知性の高低は、環境へのノイズ (突発的な攪乱) に適切に対処できる能力、と考えることができる。低知性者のノイズへの対処が低さが、

^{†9} 山岸 [12] 内では「他の相手を『信頼』する人は社会的知性が高く、相手の行動から相手への信頼度をすばやく修正できる」と議論されている。WR がこの議論に最も近い。

ES や RC の発生頻度として現れた、といえよう。

今回のモデルを 5 人 (実験 1)・6 人 (実験 2) と言った少人数で分析した理由は、最初にモデルの傾向を調べるためという側面もあったが、実行速度の制約も存在したためである。平均マッチング回数を同程度に保ったままモデルの人数を増やすと、1 ループに必要なマッチング回数が大きく増加するだけでなく、収束までに必要なループ数も増大する。現在の Mathematica 環境では高速化に限界があるので、高速な言語への移植は不可避である。

モデルの不要部分の簡素化と実行の高速化を行う事で、モデル内記述の豊富化と多人数でのシミュレーションの実現を可能にする事が、今後の課題となるだろう。

参考文献

- [1] W.Brian Arthur, J.H. Holland, B. LeBaron, and P. Tayler. Asset pricing under endogenous expectations in an artificial stock market. In *The Economy as an Evolving Complex Systems II*, pp. 15–44. Addison-Wesley Publishing, 1997.
- [2] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.
- [3] Robert Axelrod. *The Complexity of Cooperation*. Princeton University Press, Princeton, New Jersey, 1997.
- [4] 出口弘. 『複雑系としての経済学』. 日科技連, 2000.
- [5] Joshua M. Epstein and Robert Axtell. *Growing Artificial Societies*. MIT Press, Cambridge, Massachusetts, 1997.
- [6] Robert H. Frank, Thomas Gilovich, and Dennis T. Regan. Does studying economics inhibit cooperation? *Journal of Economic Perspectives*, pp. 159–71, 1993.
- [7] Richard J. Gaylord and Louis J. D'Andria. *Simulating Society*. Springer-Verlag, New York, 1998.
- [8] 松島斉. 「過去、現在、未来: 繰り返しゲームと経済学」. 伊藤元重, 岩井克人 (編), 『現代の経済理論』. 東京大学出版会, 1994.
- [9] Thomas C. Schelling. *Micromotives and Macrobehavior*. Norton, New York, 1978.
- [10] Jörgen W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, Massachusetts, 1995.
- [11] 山岸俊男. 『社会的ジレンマの仕組み』. サイエンス社, 1990.
- [12] 山岸俊男. 『信頼の構造』. 東京大学出版会, 1998.
- [13] 盛山和夫, 海野道郎 (編). 『秩序問題と社会的ジレンマ』. ハーベスト社, 1991.

エージェント系としてのレプリケータダイナミクスとその制御

國上真章 寺野隆雄

マルチエージェント系においては、協調現象や秩序の創発、知性の発現などが期待されている。しかし同時にマルチエージェント系においては、状態不安定性によるカオス的で予測不能な状態が発生する。本稿においてはこのような状態不安定性が、複数エージェント間の相互作用により普通に発現しうるものであることを、レプリケータダイナミクスによるエージェントの表現で検証すると共に、カオス制御の分野で試みられてきた手法の適用について検討したものである。

1 マルチエージェント系

エージェントには様々な解釈がありうるが、何らかの合理性に基づく独自の意思決定と行動のメカニズムを持ち、他のエージェントを含む外界（環境）と相互作用を行うものと定義される。このようなエージェントが複数、しかも動的に相互作用を行う状況においては、単一エージェントの性質からは予想されない種々の性質が観察される。

このようなマルチエージェント系については、エージェント間の相互作用による、秩序や知能の創発現象の研究がなされてきた。ここでいう秩序の創発とは、単純で均質なエージェントの集まりからの複雑な秩序の生成・分化（自己組織化）であり、また多種多様（かつ利己的）なエージェント集団からの、協調行動や社会的最適性の発生である。また知能の創発とは、自律・分散するエージェントの自発的協調または適応的学習により柔軟かつ高度の作業が可能になることである。

更に、実世界の複雑な現象を複数のエージェントの創発現象として実現する試みとして、市場メカニズム（例えば

契約ネットプロトコル）の研究や知的ロボットメカニズム（例えば人工免疫システムなど）の開発が行われてきている。

このように「望ましい」状態の創発が取り上げられる反面、創発に至らない、または創発に失敗した（おそらく無数の）系については、十分な研究が行われてきたとはいえない。しかしながら、マルチエージェント系によるシミュレーションを行ったからといって、秩序や知性が予定調和的に発現すると決まっているわけではない。（発現が難しいからこそ様々な手法が研究されるのだが、その際目的どおりの振舞いをしないエージェントについてはモデル化の失敗として省みられないことがあるのではないか）

これらのマルチエージェント系における「望ましくない」状態として、早期収束やロックインなどが扱われてきたが、混乱あるいは痙攣とでもいうべき状態不安定性も特別な仕掛け無しに発生する普通の状態であると考えられる。本稿においては、可能な限り一般的で簡単なエージェントを用いて、マルチエージェント系が本質的に不安定で予測不能な状態不安定性（カオス）を含んでいることを示す。

このような状態不安定性は、系の多様性を維持する（ホメオカオス）という重要な側面もさることながら、協調ロボットシステムなどにおいて時に危険な予測不能性や不適切な強化信号の発生を引き起こすこともあり、何らかの安定化の仕掛けを構築できることが実用上必要である。逆に状態不安定な系は上手に制御できれば、逆バタフライ効果により大きな制御ゲインを期待できる。その意味からも、マルチエージェント系という枠内での制御について検討することは意味のあることだと考えられる。

2 マルチエージェント系における状態不安定

マルチエージェント系における状態不安定についての研究としては、池上らによる、繰り返しゲームにおける戦略の進化とカオスの縁に関する研究がある[8][9]。この中では、エージェントの行動変化は変異－淘汰メカニズムに依存している。しかしながら現実の社会集団においては、複雑な(価格や取引量の)変化が、変異(新製品、新戦略)の発生－淘汰サイクルよりはるかに短いタイムスケールにおいて生起しているように見える。各エージェント内部における既知の選択枝の範囲内でも、エージェント同士の相互作用によって複雑な態度変化が駆動されるのではないかと考えられる。

他にマルチエージェント系におけるカオスとその制御に関するものが Hogg-Hurberman 系の研究で、複数の資源を競合して利用するエージェント間の時間遅れ、情報の不確かさなどに起因するカオスとその安定化を扱っている[6][10][14]。Hogg-Hurberman 系においては、時間遅れ、情報の不確かさが相互作用内の明示的な非線形関数で定義されており、カオス的な挙動の原因としてこれらの非線形性が最初から仮定されているため、エージェントとしてはやや特殊なものとなっている。

本稿においては、カオス的な挙動の普遍性を示すため、KISS 原理[1]に基づく簡単で一般的な構造を持つエージェントにおける状態不安定性の発生について以下のような検討を行う。

- ・エージェントとしての最小限の要素をもち、解析的な性質が良くわかるシステムの構築
- ・上記システムで状態不安定が起こること
- ・状態不安定が、エージェントの相互作用により発現していること
- ・状態不安定に対してカオス制御が有効に機能し、マルチエージェント系の制御として望ましい性質を持つこと

3 RD によるマルチエージェント系の表現

先の主張を検証するため、本稿ではマルチエージェント系を RD(Replicator Dynamics)を用いて表現し、数値実験をおこなう。表現は KISS 原理に基づき可能な限り簡明なも

のとするが、エージェントとして次の要素を最小限含むものとする。

- ・外部の情報の受容
- ・内部モデル(信念あるいは合理性)に基づく意思決定
- ・外部に対する行動

エージェントの表現に用いる RD は、主に進化ゲーム理論において集団内の人口構成比率の力学系として用いられ、通常以下のように記述される。

$$\frac{1}{x_i} \frac{dx_i}{dt} = c \cdot ((\mathbf{Ax})_i - \mathbf{xAx}), \quad \sum_{i=1}^N x_i = 1 \quad \cdots(1)$$

$$\mathbf{x} = (x_1, x_2, \dots, x_N), \quad \mathbf{A} \in \mathbf{M}_{N \times N}$$

進化ゲームにおいては、 x_i を集団内でとりうる N 種類の戦略のうち第 i 番目を選択する個体の人口構成比率、 i 番目と j 番目の戦略をとる個体が無作為に対戦(ランダムマッチング)した場合の利得行列を $\mathbf{A}=\{a_{ij}\}$ として解釈する。力学系(1)は、第 i 番目の戦略を選択する個体の人口比率の相対増加率が、第 i 番目の戦略による期待報酬(戦略 i の期待適応度)と全集団の平均報酬(平均適応度)との格差に比例することを要請している。この力学系については、主として進化ゲームの立場から均衡、進化的安定性などの性質が調べられてきた[5][15]。また、多触媒系や人工生命の人口動学の解析にも用いられている[3]。

また、最近これら進化ゲームとは別の立場からエージェントの挙動に関するモデルとしての解釈から社会制度設計への適用が提案されている[19]。

これと類似の力学系としては、捕食者－非捕食者の人口動学を記述する、Lotka-Volterra 系がよく調べられているが、両者は原点及び無限遠点を除き微分同相であることが知られている[5]。この Lotka-Volterra 系においてはカオスの発生する系が報告されており[2]、当然同値な RD においてもカオスが発生する。

進化ゲームにおけるように x_i を純粋戦略 i を選択する人口の比率とする描像においては、個体は特定の戦略のみしか採らず適応度に応じた選択によってその人口比率を変化させるだけの存在である。これでは、内的な合理性あるいは信念に基いて意思決定を行うエージェントのモデルとはいえない。しかしながら RD においては x_i を特定のエージェント内部における資源 i への配分比率(ゲーム論的には混合戦略)とみなすことも可能である。そこで、資源配分 $\mathbf{x}$ にお

ける選択肢 i の頻度(選好度)の相対的变化率が, x を選択したときの第 i 成分に対する評価は, 第 j 成分による第 i 成分への肯定的評価(刺激効果) $a_{0ij} x_j$ と 否定的評価(抑制効果) $-k_a a_{0ji} x_j$ の総和 $(Ax)_i$ にしたがっていると見なすことにする。ここで, k_a は, 刺激抑制の相対強度である。

$$\frac{1}{x_i} \frac{dx_i}{dt} = c \cdot ((Ax)_i - xAx) \quad \dots(2)$$

$$\sum_{i=1}^N x_i = 1, \quad A = A_0 - k_a {}^t A_0$$

ここで, 右辺第二項は刺激・抑制(肯定・否定)の平均値でありこれが選好度増減の基準となる。この描像は, Farmer によって導入された Lotka-Volterra 型の人工免疫系のネットワーク[4]において変異メカニズムを考慮しない場合を RD で書き直したものになっており, このような描像に従うと, A をエージェントの自己の選択肢に対するエージェント自身の内部的評価モデル, 力学系(1)をエージェントの内部的評価モデルに基づく態度変容の記述と見なすことができる。

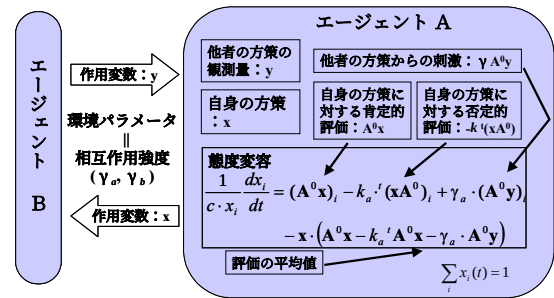
次に, エージェント同士の相互作用を導入する。最も単純な系として, 2体のエージェント A 及び B を考え, 其々3変数($x_1, x_2, x_3, y_1, y_2, y_3$)の RD で記述されるとする。ここで行列 A_0 の ij 成分 a_{0ij} は, A 系の第 i 状態に対する第 j 状態量の刺激, $-k_a a_{0ji} x_j$ は抑制を表し, 同様の関係が B_0 についても成立するとする。 A, B それぞれの状態量が行動出力となり, 他系において刺激要因として受容される(隣人のポートフォリオは魅力的に見える?)と仮定すると, 系全体の表現は次のとおりである。

$$\begin{aligned} \frac{dx_i}{dt} &= c \cdot x_i ((Ax)_i + \gamma_a (A_0 y)_i - xAx - \gamma_a xA_0 y) \\ \frac{dy_i}{dt} &= c \cdot y_i ((By)_i + \gamma_b (B_0 x)_i - yBy - \gamma_b yB_0 x) \dots(3) \\ \sum_{i=1}^3 x_i &= 1, \quad A = A_0 - k_a {}^t A_0 \\ \sum_{i=1}^3 y_i &= 1, \quad B = B_0 - k_b {}^t B_0 \end{aligned}$$

ここで, エージェント AB 間の相互作用を導入したのであるが, 相互作用の表現には他にも方法がある。一つは, それぞれの刺激抑制行列 A, B が他エージェントの状態の関数であるとみて, $A=A(x,y), B=B(y,x)$ とするやり方である。この方法は各エージェントが, 他者を外界(の一部)として観測し, それに従って自身の内部モデルを変容させるという描像のストレートな表現として魅力的である。が, 反面モデルが不

必要に高次の非線形項を含むことと内部モデルの変化メカニズムを明示的に指定しなければならないのが難点である。本稿で使用した, 他のエージェントの状態量に比例する量に刺激行列を作用させて(2)の右辺に印加する方法では, 内部モデル自体の変化は表現しにくい反面, RD としては非線形項の次数が上がらないため, 状態不安定性の最小実現と制御のプロトタイピングいう立場から有利である。前者の方法によるモデルは機会を改めて検討することとする。

このモデルの構成を図解すると下図のような, エージェント内部の刺激抑制ネットワークと, エージェント間の相互刺激ネットワークという2重ネットワークシステムとなっている。



このような2重ネットワークは, 群ロボットシステムへの応用において, 協調行動の創発が報告されているが[17][21], 実用システムとしては, カオス的で予測困難な挙動の発生予測やその制御についても検討を加える必要がある。

4 RD 型エージェント系のカオス

前項で構成した系に対して, 刺激反応行列とシステム時定数を以下の様にして相互作用パラメータ γ を変化させながら挙動を観察する。

$$A^0 = \begin{pmatrix} 0.15 & 0.05 & 0.30 \\ 0.25 & 0.15 & 0.10 \\ 0.12 & 0.50 & 0.20 \end{pmatrix}, \quad B^0 = \begin{pmatrix} 0.15 & 0.35 & 0.13 \\ 0.02 & 0.15 & 0.33 \\ 0.20 & 0.10 & 0.15 \end{pmatrix}, \quad k_a = k_b = 0.5, \quad c = 8.0$$

相互作用パラメータ γ が 0, 即ちお互いが独立している場合は, どちらの系も定常値に収束する(図1)。 γ がある程度大きくなると, 周期性が観察されるようになり, γ がある値に達すると, 系は疑周期的な状態から(図2), 不規則な挙動を呈する(図3)。時々現れる似通ったパターンに全く異なる状態が引き続くことは, 系が状態不安定となっていることを示唆しており, アトラクタに沿って最大リアプノフ指数を計算

[13]すると正の値を得ることから、系が状態不安定すなわちカオス的な状態にあることがわかる。

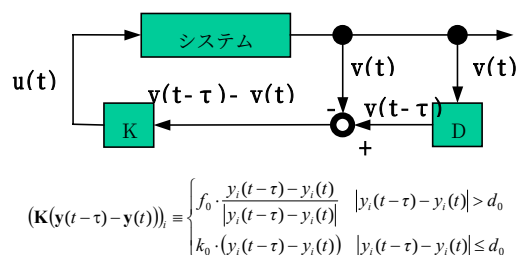
上で示した系は、2つの連続な3次元 RD 間の相互作用である。それぞれの RD は、状態変数の和が1という保存量を持つため独立な自由度は2であり、ポアンカレ・ベンディクソンの定理によって単独 ($\gamma=0$) でカオスを発生することはない。このため、先の状態不安定性は、カオス的でないエージェント複数間の相互作用によって引き起こされていると解釈できる。

この例では、相互作用の強度を表すパラメータ γ の変化により状態不安定にいたる分岐が引き起こされている。相互作用の強度は、実世界においては距離や取引費用に対応すると考えられる。このことは、エージェント間の取引費用や空間距離の変化を含むより大きな系においては、そのような値の変化が部分系においてカオスの振舞いを発生させる可能性を示しており、カオス状態の検知や制御が実用上重要であることを示唆している。

5 RD 型エージェント系のカオス

ここでは、RD で記述されるマルチエージェント系のカオスにおける制御について検討する。マルチエージェント系においては、ミクロな状態量に対してマクロな制度パラメータによりシステムの制御を行う、ミクロ・マクロリンクによる制度設計が提案されている[19]。これは状態不安定性の制御においては、システムの観測・同定に基づく分岐パラメータのチューニング (OGY 法[11]に代表される極配置問題) に対応し、例えば中央銀行の金利政策や政府による税制政策に通じるものである。一方実世界には、中央銀行の売買オペレーションや政府の財政出動のように、状態変数に直接印加するタイプの制御政策も存在する。以下においては、このようなエージェントの状態変数を部分的に観測しフィードバック信号を出力する制御が、エージェントと同程度に Simple かつ Stupid な機構によって実現しうることを示す。

連続時間系に対するカオス制御方式として、最も簡単なものとして Pyragas の遅延フィードバック制御[12]がある。この方式はカオス系を安定化したい周期と同じだけ遅延させた状態量を系にフィードバックするもので、そのブロック線図は次図に示すとおりである。



この制御方式は、ゲイン行列 K が試行錯誤的にしか定まらないこと、また安定化の目標となる周期軌道が複数存在する場合そのどれに収束するか事前にわからないこと等の短所はあるものの、観測量のみに基づく制御であること、また状態変数の一部にしか観測とフィードバックが及ばない時でも制御可能となる場合があることから、マルチエージェント系のリアルタイム制御 (安定化) 方式として検討の価値がある。

図4は、図3に示したカオス状態にある RD 系の A 系の x_2 と B 系の y_1 に対して、途中から Pyragas の遅延フィードバックを適用した結果である。状態変数の一部分に対する観測とフィードバックでありながら、系がほぼ安定化され予測可能な状態となっており、制御が有効に機能していることがわかる。

カオス制御については、すでに様々な研究が行われており[18]、典型的なものとしては離散時間システムに対する OGY 法[11]について様々な一般化や改良が研究されている。相互作用する RD 系においては差分システムにおいてもカオスが発生するので、OGY 法も制御方式として有効である。しかしながら、OGY 法は系のダイナミクスが既知であることが必要であり、エージェント達を観測情報に基づいて制御するという立場からは、まず系の同定作業が必要となる。系の同定については、カオス時系列の解析[16]から同定に必要な自由度の推定を行い、遺伝的プログラミングによりダイナミクスの同定を行う[7][20]ことが考えられる。ただし、制御対象の単純さに比較して制御側に複雑な機構を要求するため、エージェント数が増加した場合にリアルタイムな制御方式として有効かどうかは疑問である。

6 まとめ

前項自己の内部モデルと外界からの信号によって態度変容を行うエージェントの挙動を、RD によってモデル化することが出来る。これによって RD のような比較的単純な少自由度かつ小数のエージェントにおいても、(特殊な非線形関数

を含まない) 相互作用によって状態不安定性が発現することが示された。

マルチエージェント系の制御は, 系の内部モデルに対する事前知識を伴わない観測にもとづく制御である必要がある。このような制御が制御対象となるエージェントに比べて過剰に複雑となることは, 計算量の点から考えても好ましくないが, 本検討においては, 遅延フィードバックのような「覚えて遅れて出力する」だけの制御でも有効な安定化機構となり得ることが示された。

参考文献

- [1] Axelrod, R., The Complexity of Cooperation, Princeton 1999.
- [2] Arneodo, A., Couillet, P., Tresser, C., OCCURRENCE OF STRANGE ATTRACTORS IN THREE-DIMENSIONAL VOLTEERRA EQUATIONS, Physics Letters A Vol.79 No.4, 1980
- [3] Forst, C., Complex dynamics of molecular evolutionary process, in Unifying Themes in Complex Systems, Bar-Yam(eds.), Perseus Books, 2000
- [4] Frammer, J.D., Packard, N.H., Perelson, A.S., THE IMMUNE SYSTEM, ADAPTATION, AND MACHINE LEARNING, Physica D Vol.22, 1986
- [5] Hofbauer, J., Sigmund, K., Evolutionary Games and Population Dynamics, Cambridge Univ. Press, 1998
- [6] Hogg, T., Huberman, B., Controlling Chaos in Distributed Systems, IEEE Transactions on Systems, Man, and Cybernetics, Vol.21 No.6, 1991
- [7] Ikeda, Y., Tokinaga, S., Approximation of Chaotic Dynamics by Using Smaller Number of Data Based upon the Genetic Programming and Its Applications, IEICE Transactions, Fundamentals Vol.E83-A No.8, 2000
- [8] Ikegami, T., Kaneko, K., COMPUTER SYMBIOSIS-EMERGENCE OF SYMBOTIC BEHAVIOR THROUGH EVOLUTION, Physica D Vol.42, 1990
- [9] Ikegami, T., From genetic evolution to emergence of game strategies, Physica D vol.75, 1994
- [10] Kephart, J.O., Hogg, T., Huberman, B.A., Dynamics of computational Ecosystems, Physical Review A, Vol.40 No.1, 1989
- [11] Ott, E., Grebogi, C., Yorke, J.A., Controlling Chaos, Physical Review Letters, Vol.64, No.11, 1990
- [12] Pyragas, K., Continuous control of chaos by self-controlling feedback, Physics Letters A, Vol.170 No.6, 1992
- [13] Steeb, W.-H., THE NONLINEAR WORKBOOK, World Scientific Pub., 1999
- [14] Ushio, T., Imamori, T., Yamasagi, T., Controlling Chaos in Discrete-Time Computational Ecosystem, in Controlling Chaos and Bifurcations in Engineering Systems, Chen(eds.) CRC Press, 2000
- [15] Weibull, J.W., Evolutionary Game Theory, MIT Press, 1995
- [16] 合原一幸編, カオス時系列解析の基礎と応用 産業図書, 2000
- [17] 市川 真吾, 石黒章夫, 内川芳樹, 免疫処理情報機構を参考にしたロボットの知能発現, システム/制御/情報, Vol.43, 1999
- [18] 潮 俊光, カオス制御, 朝倉書店, 1996
- [19] 出口 弘, 複雑系としての経済学, 日科技連, 2000
- [20] 時永 祥三, 複雑系による経済モデル分析, 九州大学出版会, 2000
- [21] 福田 敏男, インテリジェントシステム, 昭晃堂, 2000

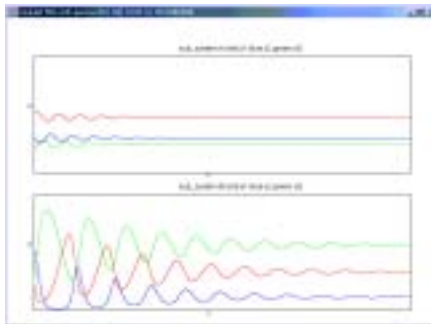


図 1：状態推移($x - t, y - t$) 及び相図($x_1 - x_2, y_1 - y_3$) (γ_a, γ_b) = (0, 0)

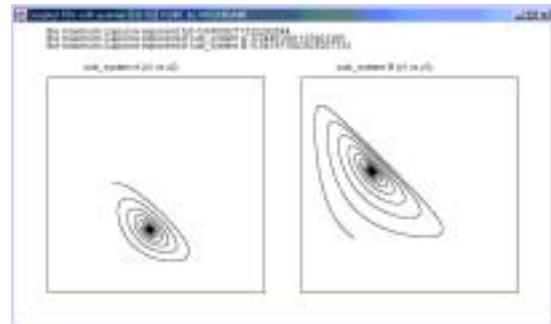


図 2：状態推移($x - t, y - t$) 及び相図($x_1 - x_2, y_1 - y_3$) (γ_a, γ_b) = (0.5, 0.15)

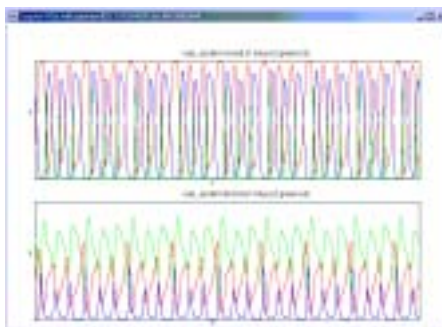


図 3：状態推移($x - t, y - t$) 及び相図($x_1 - x_2, y_1 - y_3$) (γ_a, γ_b) = (0.6, 0.15)

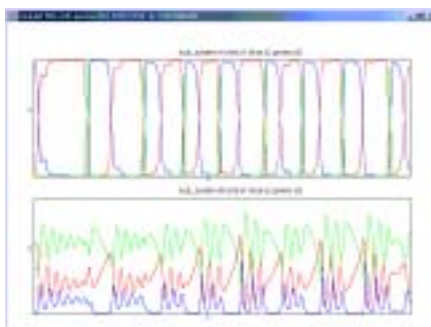
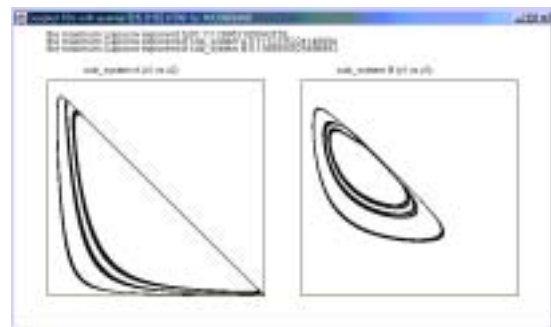
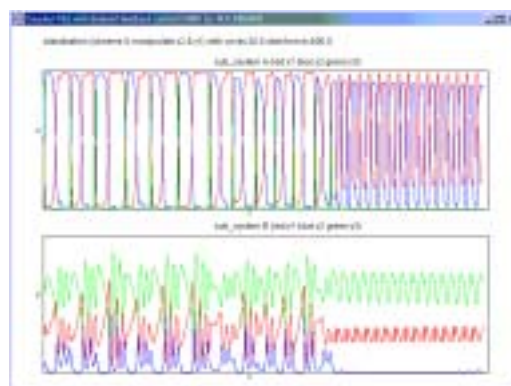


図 4：遅延フィードバック ($d_0=0.01, f_0=0.005, k_0=0.75$) を x_2 及び y_1 に適用



エージェントシミュレーションによる Prospect 理論と GARCH モデルの関連性の 分析

高橋 大志 寺野 隆雄

GARCH (Generalized Autoregressive Conditional Heteroscedasticity) モデルはマクロレベルな事象を説明するモデルであり、金融工学の分野において報告されているボラティリティを推定するモデルである。一方 Prospect 理論はミクロレベルのモデルであり、認知心理学において報告されている人間の意志決定を示した理論である。本稿は、ファイナンス、認知心理学の分野において広く知られた両理論を、エージェントベースシミュレーションにより結びつけ、ミクロレベルの挙動を基にマクロレベルの分析を行なったものである。

本稿では、4 タイプの（合理的/合理的でない）投資家からなる仮想市場を構築し、以下の事柄を確認した。(1) 合理的な投資家が存在しても価格は理論価格から乖離するケースがある、(2) 自然選択など報告が成立しないケースがある、(3) 合理的でない投資家が、GARCH モデルや非対称性を取り込んだ GARCH モデルに従う価格変動を引き起こす。これらの結果は実際の市場においても合理的でない投資家が価格に絶えず影響を与えていることを示唆するものであり、更にエージェントシミュレーションをファイナンスの分野に適用することの有効性を示すものである。

1 経緯

近年日本の人口構成の急速な高齢化が見込まれており資産運用業務の重要性はますます高まっている。

Hiroshi TAKAHASHI, 中央三井信託銀行, The Chuo Mitsui Trust and Banking
Takao TERANO, 筑波大学, University of Tsukuba

資産運用を適切におこなうため、資産価格変動のメカニズムを明らかにすることは極めて重要性が高い。

ファイナンスの分野において資産価格を導出するモデルとして広く知られている Capital Asset Pricing Model(以下 CAPM と呼ぶ) は、マーケット・個人の行動に一定の仮定をおき、解析的手法により均衡資産価格を導くモデルである。例えば投資家については von Neumann and Morgenstern の期待効用最大化に基づき意思決定を行う合理的投資家を仮定し、市場については税金等のない完全な市場を仮定している。

CAPM をはじめとする従来のファイナンス理論のモデルは、「仮定が非現実的である」、「モデルでは説明できないアノマリーと呼ばれる現象が存在する」など数多くの問題点が指摘されている。それらの現象を説明するいくつかの仮説が報告されているが、マーケット価格の挙動を説明する決定的な理論は報告されてはいない。

2 目的

本稿の目的は、行動ファイナンスに基づく投資家および投資制度が資産価格に与える影響をエージェント・ベース・シミュレーションにより分析し、従来ファイナンスや金融工学の分野にて報告されている以下の事柄を検証することである。

(1) 効率的市場仮説によれば、「一握りの合理的な投資家が存在すれば合理的な価格付けがなされる」とされているが、これは常に成立するのか？

(2) Friedman[8] は、自然選択の原理により合理的でない投資家は排除され、合理的な投資家は生き残ると

の報告をしているが、従来の理論とは反対に合理的な投資家が淘汰されてしまう状況はおこらないのか？

(3) 合理的ではない投資家が、金融工学の分野において報告されている GARCH モデルに従う価格変動を引き起こすのではないだろうか？

3 関連研究

3.1 合理的な意思決定について

従来のファイナンス理論は、投資家が期待効用最大化に基づく合理的な意思決定を行うことを仮定した上で、資産価格を導出するモデルを構築している。Ingersoll[11] は投資家の効用関数の形状について、いくつかのタイプを想定した上で、資産価格や多期間最適消費戦略の導出などをおこなっている。

時間の経過と共に新しい情報が到達する場合、対象事象の事前確率分布と新規情報を基に事後的な確率分布を推定する必要がある。Howson[10] らは、このような信念の更新を行うケースにおいては、ベイズに基づく方法が合理的な意思決定の基準であるとの報告を行っている。ファイナンスの分野においてベイズを利用した例としては、Black [4] にて報告されたモデル^{†1}が挙げられる。当モデルは、株式が内在的に含む均衡収益率と短期的な期待収益率からベイズ修正により統合収益率を算出した上で、リスクとリターンを考慮した最適な資産配分を算出するモデルである。当モデルは、合理的な投資の意思決定プロセスとして、実務においても広く使われている。

3.2 人間の意思決定について

認知心理学の分野において Kahneman[13] らは、人間の意思決定が期待効用最大化から乖離する傾向のあることを報告している。彼らは期待効用理論にかわる理論を提示し、これはプロスペクト理論と呼ばれ、人間の意思決定に関し、(1) 人間の意思決定は参照点からの変化に基づき行なわれる、(2) 損失の受け取り方は利得と比較して大きい^{†2}、などの特徴があることを示唆している [12]。

^{†1} 以下、ベイズ修正モデルとよぶ。

^{†2} 例えば、10 万円の利得と 10 万円の損失を比べた場合、損失の方を大きく見積もる傾向がある、など。

意思決定の合理性からの乖離に関しては、人間は自分の意思決定や能力に対して過剰の自信を持つ Over Confidence や、情報を受け取ったときにベイズと比較して小さすぎる信念の更新しか行わない過小反応などが報告されている [3][14]。

3.3 予測における合理性からの乖離について

従来のファイナンス理論 [7] では、ファンダメンタルズから理論価格を導くことができるとの考え方が一般的であるしかし Shiller [15] は、それとは反対に”実際のマーケットでは現時点の株価を正当化するように新しい理論が作りだされている”、との指摘をしている。このようなメカニズムが働く状況においては直近の株価の推移が将来の価格に与える影響は大きい。

また、社会心理学の分野から同じ情報をみても自分に都合良く情報を受け取ってしまうセルフ・サービング・バイアスの存在が報告されており、このバイアスのため将来の予測については投資家毎にばらつきが生じると考えられる。

本稿においては、完全な情報を有し期待効用最大化に基づく意思決定を行なう投資家を合理的な投資家とし、それ以外の投資家を合理駅ではない投資家と呼ぶこととする。合理的でない投資家は、H.A.Simon のいう限定合理性と同様のものである。

3.4 人工市場を用いた分析

近年人工知能の研究者を中心としてコンピュータ上に人工市場を構築し、シミュレーションにより市場を分析する手法が行なわれるようになってきた。

Arthur[2] らは仮想市場内に適応的に投資行動を変化させる異質な投資家を設定し、価格変動の分析をおこなっており、その中で資産価格のバブルやクラッシュが生じることや、理論価格から取引価格が乖離する現象は頑健に生じるとの報告をおこなっている。

和泉 [16] らは、投資銀行の為替ディーラーの行動の観察を基に外為市場のマルチエージェントモデルを構築し、マクロ経済指標などと絡めた分析をおこなっている。その中で、正のフィードバック効果により市場がフラット相からバブル相に相転移することや、ほとんど全てのエージェントが同じ予想をするようにな

るとトレンドが消滅しコントラリーオピニオン現象がみられることなどの報告している。

Epstein[6] は、エージェントが市場内を移動しながら局所的に取引を行う市場による分析をおこなっており、その中で「市場機構という仕掛けに任せておきさえすれば自動的に均衡価格がもたらされる」という自由放任主義的な経済政策の理論的立場は苦しくなるとの報告をしている。

また、出口 [17] らは、仮想株式先物市場 U-mart を構築し、マシンエージェントと人間の共存する人工市場による分析をおこなっている。

従来の人工市場に基づく分析の多くは、エージェントの行動を KISS 原理 [1] に従い記述しているが、本稿では投資家の意思決定をより現実に近いものとしており、その意味で KISS 原理とは異なるものである。

3.5 価格変動の分析

ボラティリティに関しては、Shiller が資産価格変動がファンダメンタルズでは説明できない程大きいことを指摘したのをはじめとし、株価変動の大きい時期は継続する傾向があることや、株価が下がった後と上がった後における価格変動が非対称であるとの報告など、数多くの報告がなされている。

金融工学の分野において、資産価格のボラティリティを推定するモデルは数多く報告されており、最も簡単な (1) 最小自乗法 (OLS) によるものや、(2) ボラティリティに自己相関のあることを取り込んだ GARCH モデル [5]、(3) ボラティリティ変動の非対称性を取り込んだモデル [9] などのモデルが挙げられる。

$$\varepsilon_t \sim N(0, \sigma^2), \text{ where } \sigma^2 \text{ is constant,} \quad (1)$$

$$\sigma_t^2 = a + \sum_{i=1}^p a_i \varepsilon_{t-i}^2 + \sum_{i=1}^q b_i \sigma_{t-i}^2, \quad (2)$$

$$\sigma_t^2 = a + \sum_{i=1}^p (a_i \varepsilon_{t-i}^2 + \gamma_i S_{t-i} \varepsilon_{t-i}^2) + \sum_{i=1}^q \sigma_{t-i}^2$$

$$S_{t-i} = \begin{cases} 1 & \text{if } \varepsilon_{t-i} < 0 \\ 0 & \text{if } \varepsilon_{t-i} \geq 0 \end{cases} \quad (3)$$

上記の 3 つのモデルを実際の市場価格データ (MSCI 株式指数、Dow Jones Industry 採用個別銘柄: 日次 1980/1~2001/10、5697 日) に対しあてはめた場合の AIC^{†3} の値を示す。背景が塗りつぶされているのは、各サンプル毎に、AIC にて 3 つの中で最適と判断されたものである。表よりボラティリティの非対称性を取り込んだ GARCH モデルの当てはまりのよいことが確認できる。

表 1 AIC 基準 (MSCI 株式指数)

	AUSTRIA	AUSTRIA	BELGIUM	CANADA	DENMARK	FRANCE	GERMANY	HONG KONG	ITALY
OLS	18783	17765	17195	16558	17785	18617	19120	23476	20657
GARCH	17877	16823	16507	14681	17310	17891	18267	21436	19814
ASYMMETRY	17816	16815	16488	14645	17311	17848	18223	21279	19804

	JAPAN	NETHERLANDS	NORWAY	SINGAPORE	SPAIN	SWEDEN	SWITZERLAND	UK	USA
OLS	19785	17896	20400	19797	18927	20224	17277	17734	16324
GARCH	16716	16686	19387	17631	18024	18967	16521	17030	14911
ASYMMETRY	16665	16648	19358	17590	18019	18945	16444	17024	14823

表 2 AIC 基準 (ダウ・ジョーンズ: 個別銘柄)

	ALDOA	AMEREX	AT&T	BOEING	CATERPILLAR	COCA	DISNEY	DJ PONT	EASTMAN	EXXON
OLS	23788	24780	23819	23887	23838	23015	23370	22231	22007	20281
GARCH	23042	24008	21234	23121	23182	21100	23011	21648	22223	19364
ASYMMETRY	23022	23958	21234	23080	23141	21082	22965	21628	22191	19348

	GENELEC	GENERAL	HENLETT	INTEL	INTL BUS	INTL PAP	JOHNSON	JPMORG	MCDONA	MERCK
OLS	21288	22980	26176	27728	22859	23243	21699	24524	21745	21423
GARCH	20301	22487	25850	27142	21804	22369	21178	23640	21669	20945
ASYMMETRY	20254	22448	25843	27104	21822	22365	21130	23582	21657	20917

	MINNESOTA	PHILIP	PROCTER	UNITED	WAL
OLS	20804	22797	21879	22528	23849
GARCH	19800	22126	20514	21981	23297
ASYMMETRY	19815	22127	20554	21986	23285

4 モデル

本稿では、従来のファイナンス理論の仮定を拡張した分析を行う。分析に用いる、市場の構成要素、投資家の取引ルールを以下に示す。

4.1 市場の構成要素

本市場は、1000 人の投資家からなるコンピュータ上の仮想市場であり、株式と無しリスク資産の 2 資産が取引可能である。利益はブラウン運動に従い毎期発生し^{†4}、各投資家は利益、過去の価格情報などを基に取引をおこなうものとする。

^{†3} Akaike Information Criteria

^{†4} 初期値は 0.5 とした。

表 3 仮想市場の構成要素

投資家の数	1000人	—
資産 (2種類)	株式(1000株) 無リスク資産	Arthur[1]と同様2種類
利益変動	ブラウン運動	Shleifer[13]と同様
投資家のタイプ (4タイプ)	合理的投資家 プロスペクト理論 Over Confidence トレンド予測	配当割引モデル 参照点からの損失を2倍大きく見積もる ボラティリティを小さく見積もる 過去のトレンドを外挿(直近10日間)

```

forecastshort (価格0)           %予想リターン
{ 価格0 = profit0 / δ + profit0    %予想価格+利益
  forecastshort = 価格0 / 価格0-1    %予想リターン }
σ0 { σ0 = 0.2 / 2000.5 }        %予測ボラティリティ(一定)
% profit0: t 期に発生した利益(予測時点で既知)
% δ : 株式の要求収益率 (共通・一定: 10%/200days)

```

図 2 合理的投資家の予測

```

%Input Investor's View
equilibrium Return = 資産配分に Implied された均衡収益率(全投資家共通)
(均衡収益率 = 2.1 * σ02 * wstock * rr, wstock: 市場時価総額における株式の割合)
σ0 = 長期の期待収益率の予想標準偏差 (共通・一定: 1.0 * 10-3)
forecastlong (= 短期の期待収益率 (タイプ毎に異なる関数: 後述))
σ2 (= 短期の期待収益率の予想標準偏差 (タイプ毎に異なる関数: 後述))
λ = 投資家のリスク回避度 (共通・一定: 1.2)
δ = 株式の要求収益率 (共通・一定: 10%/200days)
rr = リスクフリーレート (共通・一定: 5%/200days)
fund = 各投資家の総資産額(初期値: 2000 共通)

%Input Lower bound and Upper bound
Lower bound = 投資比率の下限
Upper bound = 投資比率の上限

%Integrate Investor's views
Integrate Return =  $\frac{1}{\sigma_0} \text{equilibrium Return} + \frac{1}{\sigma_0} \text{forecast}_{long}()$ 

$$\frac{1}{\sigma_0} + \frac{1}{\sigma_0}$$

% Solve the optimization problem to calculate optimal stock weight (w)
% Max: (1 + Integrate Return) * w + (1 + m) * (1 - w) * λ * w * σ22 * w
% s.t. w ≤ upperbound
% w ≥ lowerbound
%

$$w = \frac{\text{Integrate Return} - r_f}{2 \cdot \lambda \cdot \sigma_2^2}$$

if w > upper bound then
  w = upperbound
else if w < lower bound
  w = lowerbound
end if

```

図 1 意思決定アルゴリズム (共通)

4.2 投資家の意思決定

全投資家は、長期期待収益率に基づく資産配分 (以下 SAA と呼ぶ) を基準とし、短期期待収益率を反映した資産配分調整をおこなう^{†5} (図 1 参照)。本稿では、長期期待収益率を全投資家共通としたことから、SAA も全投資家共通となる。市場価格は、需要と供給が一致する価格にて決定される。

以下各投資家の短期予測方式について説明する。

4.2.1 合理的投資家の予測

配当割引モデルに基づく意思決定を合理的投資家の予測とした。本稿では、利益が每期公表されること

```

%プロスペクト理論の予測リターン及びボラティリティ
forecastshort (価格0)           %予想リターン
{ 価格0 = profit0 / δ + profit0    %予想価格+利益
  if (価格0 < 参照点価格) & (価格0 < 参照点価格) then
    価格0 = 参照点価格 - 2.25 * (参照点価格 - 価格0)
  end if
  forecastshort = (価格0 / 価格0-1) * (1 + ε) % ε ~ N(0, 0.01)
  σ2 { σ2 = 0.2 / 2000.5 }        % ε ~ N(0, 0.01)
%参照点価格: 過去の株式取引価格(本稿では 2 日前 or 5 日前の価格)

```

図 3 プロスペクト理論に基づく予測

```

forecastshort (価格0)           %予想リターン
{ 価格0 = profit0 / δ + profit0    %予想価格+利益
  forecastshort = (価格0 / 価格0-1) * (1 + ε) % ε ~ N(0, 0.01)
  σ2 { k = 0.2; σ2 = k * 0.2 / 2000.5 } % k: 自信過剰の程度を示す

```

図 4 OverConfident な投資家の予測

から、公表利益を基に価格が予測される。

4.2.2 プロスペクト理論に基づく投資家の予測

本分析では、損失の受け取り方が利得に対し 2 倍程度大きいことをモデル化した。具体的には、直近価格および予想価格が参照価格よりも低い場合に損失を大きく見積もる。株価予測は、合理的投資家と同様に配当割引モデルによるものとする。

4.2.3 OverConfident な投資家の予測

本分析では、OverConfidence を株式のリスクを小さく見積もる投資行動とした^{†6}。

4.2.4 トrend予測をする投資家の予測

トレンド予測は直近 10 日間の株価のトレンドを外挿して予測を行う。

5 結果

まずはじめに、「一部でも合理的な投資家が存在すれば合理的な価格付けがなされる」や、「合理的でな

^{†5} ベイズ修正モデルに基づく意思決定。長期期待収益率に基づく資産配分 (SAA) を基準とし、市場環境に合わせ短期的に資産配分調整を行う運用手法は、機関投資家などはじめ広く採用されている。

^{†6} 例えば、株式に正の超過収益を見込む場合オーバーコンフィデンスな投資家は合理的な投資家と比較し多額の株式投資を行う。

$forecast_{next}(\text{価格})$ %予想リターン
 $trend = \text{最近 } 10 \text{ 日間の株価リターンの平均}$
 $価格_{t+1} = 価格_t + trend + profit_t * trend$ %予想価格+利益
 $forecast_{next} = (価格_{t+1} / 価格_t) * (1 + \varepsilon)$ % $\varepsilon \sim N(0, 0.01)$
 $\sigma_2(0) \{ \sigma_2 = 0.2 / 200^{0.5} \}$

図5 トrend予測の投資家の予測

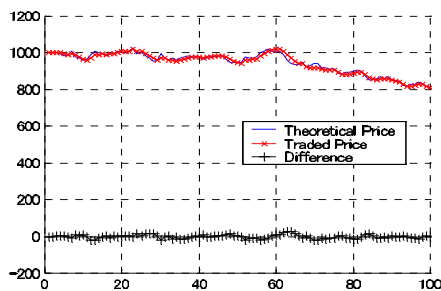


図6 価格推移 (合理的: トrend=5:5)

い投資家は自然選択の原理により排除されてしまう」などの報告が成立するかという点について、検証をおこなった。なお、分析に先立ち全ての投資家が合理的である場合に、理論価格と取引価格が一致することを確認しており、その意味でこのモデルは妥当である。また、各シミュレーションは20回繰り返し実施しており、結果の再現性は確認している。

本稿では、投資家の予測タイプの割合や、投資制約などを変化させたうえで、価格変動の分析を行った。以下各設定毎に分析結果を記す。

5.1 トrend予測の影響

合理的な投資家と、トrend予測を持つ投資家の割合を変化させて理論価格と取引価格の関係を分析した^{†7}。

5.1.1 合理的予測: トrend予測=5:5の場合

この条件では従来のファイナンス理論の報告と整合的な結果が得られた。取引価格は理論価格とほぼ一致しており、合理的な投資家が超過収益を獲得している。この状況において自然選択を働かせると合理的でない投資家が排除される可能性が高い。

^{†7} 本稿では、理論価格・取引価格及びSAAに対する投資家(一部)の超過収益を表示

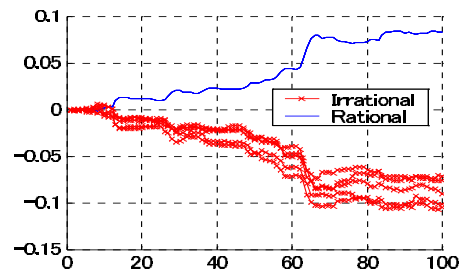


図7 累積超過収益 (合理的: トrend=5:5)

5.1.2 合理的予測: トrend予測=3:7の場合

合理的でない投資家が多い場合でも、若干の価格の乖離はみられるものの、先程と同様に従来のファイナンス理論の報告と整合的な結果を得ることができた。

5.1.3 合理的予測: トrend予測=1:9の場合

合理的な投資家の割合を極端に減らしたところ、従来のファイナンスの報告とは異なる挙動が確認できた。このケースにおいては、取引価格の乖離するスピードが、理論価格に戻るスピードを上回るため、価格が大幅に理論価格から乖離する。また、このケースでは合理的でない投資家がプラスの超過収益を獲得しており、合理的な投資家が自然淘汰されてしまう可能性が高い。

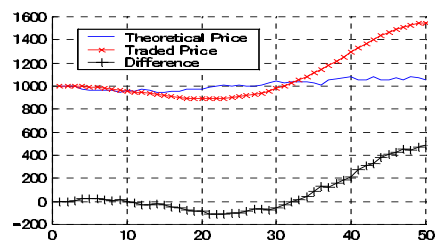


図8 価格推移 (合理的: トrend=1:9)

5.2 投資制約のある場合

合理的予測: トrend予測=3:7にて、投資制約^{†8}を加えた場合の価格の挙動を分析したところ、本

^{†8} 投資制約とは、各資産への投資割合に制約があることをさす。例: 総資金総額の50%±5%の範囲で株式に投資する、など。

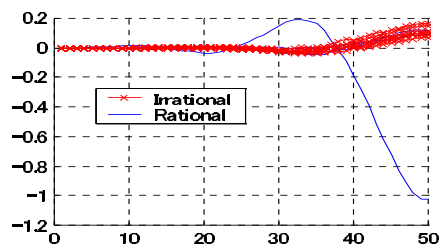


図9 累積超過収益 (合理的: トレンド=1:9)

ケースにおいても従来のファイナンスの報告とは異なる挙動を確認できた。

制約のある場合、たとえ投資家が合理的な予測を行っていたとしても、価格に対する影響が限定的となり、取引価格が理論価格から大幅に乖離する現象がみられる。またこのケースにおいても、価格支配力をもった合理的でない投資家が超過収益を獲得しており、自然選択の原理が働くと合理的な投資家が排除される可能性が高い。

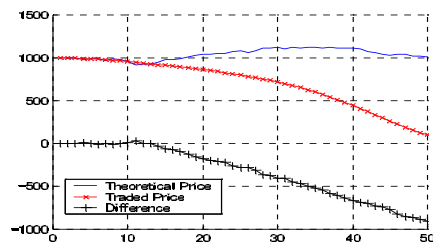


図10 価格推移 (投資制約有)

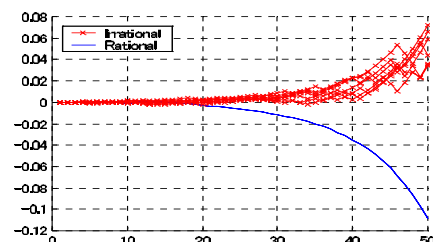


図11 累積超過収益 (投資制約有)

実際の市場においても投資制約の範囲内にて、資産運用をおこなう投資家は数多くみられるが、状況によってはそのような制約が資産価格が理論価格から乖離するひとつの要因となっている可能性がある。

更に、近年リスク管理の重要性が認識され、各金融機関とも市場リスクをはじめ、信用リスク、など各種リスク指標を用いてリスクを管理している。リスク管理の厳密化は、各個別企業のリスクを限定化する意味では有効に機能すると考えられる。しかし、リスク管理には実質的に投資制約を加える効果もあるため、副作用として取引価格を理論価格に引き戻す力を弱めている懸念がある。特に、理論価格が下落基調にある場合、リスク管理は厳密化される傾向があり、取引価格の下落とリスク管理の強化に正のフィードバックがかかり、理論価格からの乖離が大きくなる可能性がある。

このケースにおいて、合理的な投資家のうち10人(全体の1%)だけ投資制約無で投資させたところ、理論価格からの乖離は大幅に低下し、更に合理的投資家が超過収益を獲得できるようになった。

実際の市場においても、合理的な投資をおこなう投資制約の少ない投資家や資金規模の大きい投資家が一部だけでも存在することにより、理論価格からの乖離が大幅に削減する可能性がある。

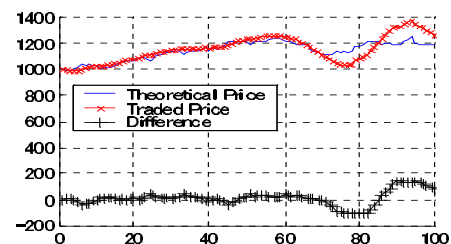


図12 価格推移 (一部投資制約無)

5.3 オーバーコンフィデンスな投資家の影響

合理的投資家: トレンド予測=5:5の割合にて、トレンド予測の投資家がオーバーコンフィデンスな場合においても、取引価格が理論価格から大幅に乖離し、更にオーバーコンフィデンスな投資家が超過収益

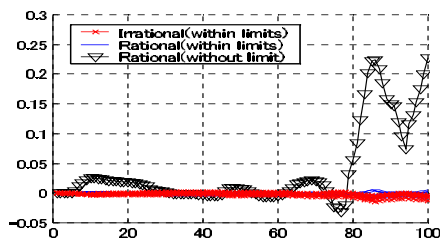


図 13 累積超過収益 (一部投資制約無)

を獲得する現象がみられた。

オーバー・コンフィデンスな投資家は合理的投資家と比較して投資割合が大きくなる為、価格に対する影響が大きいことが主な要因と考えられる。

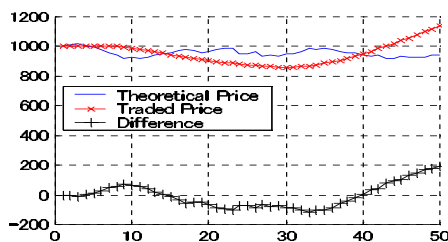


図 14 価格推移 (オーバーコンフィデンス)

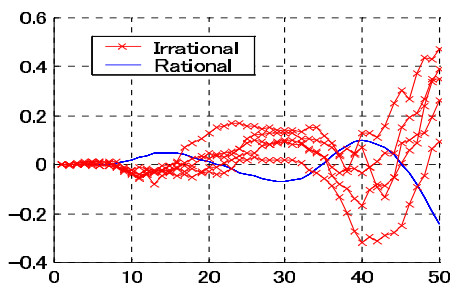


図 15 累積超過収益 (オーバーコンフィデンス)

5.4 プロスペクト理論に基づく投資家

合理的投資家：プロスペクト理論＝5：5の割合にて、プロスペクト理論の投資家が存在する場合の価格

の挙動を分析した^{†9}。

このケースにおいても、取引価格が理論価格から乖離する現象がみられる。特に理論価格が下落している場合に、取引価格は理論価格に対し下ぶれしていることを確認できる。本ケースも含めた一連の簡単な例

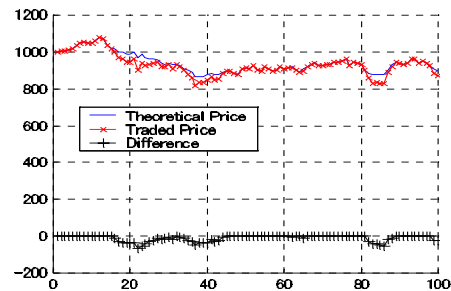


図 16 価格推移 (プロスペクト理論)

から、意思決定の合理性からの乖離は、取引価格を理論価格から乖離させ、場合によっては、合理的な投資家を排除する働きを持つことを確認できる。

6 ボラティリティの分析

本章では、「合理的ではない投資家が、金融工学の分野にて報告されている GARCH モデルに従う価格変動を引き起こすか？」という点を確認するため、シミュレーションにて得られた価格系列データに対し、時系列分析をおこなった。

6.1 データ

本分析は実際の市場価格とシミュレーション価格^{†10}を用いて分析を行った。具体的には以下のデータを対象とした。

- 合理的投資家のみ
- ノイズ予測と合理的投資家が同数
- Trend 予測の投資家と合理的投資家が同数
- Prospect 理論 と合理的投資家が同数

^{†9} プロスペクト理論の参照点は、10 日前。

^{†10} シミュレーション期間は 2500 期間とし、10 回繰り返した。また、プロスペクト理論の参照点は、2 日前と 10 日目の 2 通り。

6.2 推定結果

シミュレーション結果及び実際のマーケットデータへ各モデルをあてはめた場合の AIC の値を示す。背景が塗りつぶされているものは、各サンプル毎に、AIC にて 3 つの中で最適と判断されたものである。

表 4 AIC 基準 (シミュレーション)

Simulation Number		1	2	3	4	5	6	7	8	9	10
rational	OLS	8761	8909	8816	8808	8720	8971	8823	8686	8873	8725
	GARCH	8765	8909	8819	8809	8722	8971	8821	8688	8877	8728
	ASYMMETRY	8765	8908	8820	8810	8724	8973	8823	8684	8878	8730
noise50%	OLS	8704	8886	8796	8782	8701	8782	8677	8795	8640	8737
	GARCH	8708	8889	8798	8780	8703	8780	8678	8796	8641	8739
	ASYMMETRY	8705	8885	8800	8782	8704	8782	8680	8799	8642	8740
trend50%	OLS	6589	6396	6699	6323	6688	6444	6433	6495	6642	6601
	GARCH	6347	6168	6447	6151	6474	6253	6192	6292	6461	6427
	ASYMMETRY	6348	6170	6449	6153	6476	6254	6193	6293	6463	6428
Prospect 2days	OLS	10653	10594	10781	10636	10628	10767	10763	10768	10724	10743
	GARCH	10530	10504	10694	10552	10727	10693	10777	10629	10597	10692
	ASYMMETRY	10485	10464	10634	10508	10664	10626	10726	10580	10557	10657
Prospect 10days	OLS	10665	10606	10635	10664	10664	10722	10656	10409	10462	10488
	GARCH	10332	10644	10674	10591	10512	10642	10526	10517	10609	10434
	ASYMMETRY	10179	10396	10616	10426	10582	10453	10556	10162	10264	10280

合理的な投資家が 100%の場合および、ノイズトレーダーが存在する場合については、従来のファイナンスの想定通り OLS が選択された。

一方トレンド予測の投資家が存在する価格系列、プロスペクト理論に基づく投資家が存在する価格系列については、それぞれ GARCH モデル、非対称性を取り込んだ GARCH モデルが選択されており、合理的でない投資家が GARCH モデルに従う価格変動を引き起こすことを確認できる。

市場データ分析においても非対称性を取り込んだ GARCH モデルが選択されており、これらの結果は、実際のマーケット参加者がプロスペクト理論の意思決定と同様に利得発生時と損失発生時において異なる意思決定方式をもつことを示唆するものと考えられる

7 まとめ

エージェント・ベース・アプローチを通じ、行動ファイナンスと従来のファイナンス理論、金融工学を結び付けた分析を実施し、当初の 3 つの質問に答えることができた。すなわち、(1) 合理的な投資家が存在しても価格は理論価格から乖離するケースのあること、(2) 自然選択など報告が成立しないケースのあること、(3) 非合理的な投資家が、GARCH モデルや非対称性を取り込んだ GARCH モデルに従う価格変動

を引き起こすこと、の 3 点である。これらの結果は、実際の市場においても非合理的な投資家が排除されず、取引価格に対し絶えず影響を与え続けている可能性を示唆するものである。

参考文献

- [1] Axelrod, R. The Complexity of Cooperation - Agent-Based Model of Competition and Collaboration, Princeton University Press (1997)
- [2] Arthur, W. B. et al. Asset Pricing under Endogenous Expectations in an Artificial Stock Market. The Economy as an Evolving Complex System II, Addison-Wesley (1997), 15-44.
- [3] Bazerman, M. Judgment in Managerial Decision Making. John Wiley & Sons, 1998.
- [4] Black, R., and Litterman, R. Global Portfolio Optimization. Financial Analysts Journal, (September-October, 1992), 28-43.
- [5] Bollerslev, T. Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics, 31 (1986), 307-327.
- [6] Epstein, J. M., and Axtell, R. Growing Artificial Societies Social Science From The Bottom Up. MIT Press, 1996.
- [7] Fama, E. Efficient Capital Markets: A Review of Theory and Empirical Work. Journal of Finance, 25 (1970), 383-417.
- [8] Friedman, M: Essays in Positive Economics. University of Chicago Press, 1953.
- [9] Glosten, L. R., Jagannathan, R., and Runkle, D. On the Relation between the Expected Value and the Volatility of Nominal Excess Returns on Stocks. Journal of Finance, 48 (1993), 1779-1801.
- [10] Howson, C. and Urbach, P. Scientific Reasoning. Open Court Publishing Company, 1993.
- [11] Ingersoll, J. E. Theory of Financial Decision Making. Rowman & Littlefield, 1987.
- [12] Kagel, J. H., and Roth, A. E. Roth, (eds.). The Handbook of Experimental Economics. Princeton University Press, 1995.
- [13] Kahneman, D., and Tversky, A. Prospect Theory of Decisions under Risk. Econometrica, 47(1979) 263-291.
- [14] Shleifer, A. Inefficient Markets. Oxford University Press, 2000.
- [15] Shiller, R. J. Irrational Exuberance. Princeton University Press, 2000.
- [16] 和泉 潔, 植田 一博, コンピュータの中の市場: 認知機構を持つエージェントからなる人工市場の構築とその評価. 認知科学, 6,(1999), 31-43.
- [17] 出口 弘, 和泉 潔, 塩沢由典, 高安 秀樹, 寺野 隆雄, 佐藤 浩, 喜多 一, 人工市場を研究する社会的および学問的意義. 人工知能, 15, (2000), 982-990.

複数財オークションについて

松井知己 渡辺隆裕

1 はじめに

オークションは, Vickley [8] を契機としてゲーム理論や経済学などで盛んに研究されてきた分野であるが, オークションが電子商取引の代表的な取引モデルとして大きく発展した事から, これらの理論の応用可能性は広がった. Yahoo や e-Bay に代表されるような消費者間取引 (C2C) における電子オークションに対し, 近年は企業間取引 (B2B) の取引モデルとしてのオークションが注目されている. この分野における重要な研究課題として複数財のオークションがある.

複数財のオークションは売り手が複数の財を売りに出すオークションである. 花市場や国債の市場など複数財のオークションは昔から存在したが, これらは同種複数財 (multi-items) が売り出されるものである. これに対し, 最近注目されつつあるのは異種複数財 (multi-objects) のオークションである. 異種複数財のオークションでは, 販売される財が「組み合わせることによって, 個々の価値の総和よりも大きな価値を生む」という補完性 (シナジー効果) を持っている場合や, その逆の代替性を持っている場合がある. このような場合には, 個々の財を独立して売る事が必ずしも効率的な資源配分につながらず, オークションがう

まく機能しない場合がある. 補完性や代替性がある場合に異種複数財オークションをどのように行うかは大きな課題である.

異種複数財のオークションの応用範囲は広い. de Vries and Vohra [4] によると, 「いくつかのロジスティクスコンサルタント, 例えば SAITECH-INC のシステム SBID などは, 組合せオークションのソフトウェアを実装し, Logistics.com のシステム OptiBidTM は 50 億ドル以上の運送契約が Bid されたと主張している (2000 年 1 月)」などと伝え, 物流のプライシングでこの分野が注目されていることを主張している. また公共部門では規制緩和と市場経済重視の流れから, 電波 (周波数帯) をオークションで売り出すことが検討され, 既にアメリカやヨーロッパなどで実施されている. 周波数帯などは隣り合う周波数帯を組み合わせると, 価値が大きく増大するため, 複数の財をどのように組み合わせてオークションするかが大きな課題となっている. 今後は公的資産や公有地の売却などに対しても, 異種複数財のオークションが注目されるであろう.

複数財オークションの 1 つである組合せオークション (combinatorial auctions) は, このような異種複数財のオークションの代表的なモデルであり, 多くの研究が行われている (de Vries and Vohra [4] など). 組合せオークションは, 参加者が財のすべての部分集合に対する価値を売り手に提出し, 売り手がその情報をもとに利益が最大となるような財の配分方法を決定するというオークションである.

組合せオークションでは, 参加者は財の部分集合

Notes on multi-object auctions

Tomomi MATSUI, 東京大学大学院 情報理工学系研究科, Graduate School of Information Science and Technology, University of Tokyo.

Takahiro WATANABE, 岩手県立大学 総合政策学部 総合政策学科, Department of Policy Studies, Iwate Prefectural University.

すべてについて金額を申告しなければならないため、その申告数は指数的に増大し、財の数が大きくなった場合には現実的ではない。これに対し、筆者たちは Matsui and Watanabe [6][7] において「その集合の中のどれか 1 つが欠けても価値がなく、またその集合以外の財にも価値がないような財の集合」である「必要十分財 (necessary bundle)」が各参加者に 1 つある場合を分析した。そして、参加者が必要十分財を欲しているような状況で、参加者が「彼の必要十分財 1 つとその価値」を入札するようなオークションについて分析を行った。その結果として、必要十分財のオークションは、戦略的な行動によっても効率的な資源配分を達成できるオークションであることを示した。更に周波数オークションなどに応用できるような consecutive one property と呼ばれる性質を満たすならば、売り手の利益を最大にするような財の最適な割り当てを求める多項式時間アルゴリズムが存在し、売り手の利益を最大にするような最適解が複数ある場合（同点となる参加者群が複数ある場合）に、等確率で 1 つの解を選び出す多項式時間アルゴリズムが存在することなどを示している。本論文は、筆者たちの研究 Matsui and Watanabe [6][7] の研究成果を要約し、紹介することを目的としている。

オークションは学際的な研究分野であり、マルチエージェント理論、経済学・ゲーム理論、組合せ理論、経営学・経営情報、暗号理論などに離散して研究者が存在している。各分野の研究者が連携を取り合い研究を進めるうえで、本稿がきっかけになれば幸いである。

2 必要十分財のモデルとその応用

本稿で扱うモデルでは、一人の**競売人**（売り手）が m 個の財をオークションに出しており、 n 人のオークション**参加者**（買い手）がいる状況を扱う。参加者の集合を $N = \{1, 2, \dots, n\}$ とし、財の集合を $M = \{1, 2, \dots, m\}$ とする。財の部分集合 $S \subseteq M$ に対する各参加者 i の価値を $V_i(S)$ で表す。 $V_i(S) \geq 0$ とし、便宜的に $V_i(\emptyset) \equiv 0$ とする。

各参加者は**必要十分財**と呼ばれる財の部分集合 1 つを欲していると仮定する。各参加者にとっては、その部分集合中の財のどれか 1 つが欠けると価値がなく、

またそれ以外の財にも価値がないとする。現実にはこのような仮定がいつも成り立つとは限らない。例えば 1 つや 2 つの財が欠けても残りの財の価値は減ずることはあれ 0 になることはない場合も多いであろう。この仮定は「必要十分財中のどれか 1 つが欠けると価値が大幅に減じられ、それ以外の財にはほとんど価値がない」ような状況を近似的に表していると解釈できる。参加者 i の必要十分財を T_i とし、彼の T_i に対する価値を $v_i > 0$ とする。したがって、 $V_i(S)$ は以下のように表せる。

$$V_i(S) = \begin{cases} v_i & (T_i \subseteq S), \\ 0 & (\text{その他}). \end{cases}$$

以下では δ を財の価値の最小単位とする。すなわち、すべての $i \in N$ に対し $v_i \in \{\delta, 2\delta, 3\delta, \dots\}$ とする。

3 必要十分財のオークション

必要十分財のオークションルールは以下のような 3 ステップからなっている。

ステップ 1 各参加者 i は、自分の必要十分財 B_i とその**入札額**（落札時の購入額） b_i を同時に提出する（各参加者は正直に入札するとは限らないので、 (B_i, b_i) と (T_i, v_i) を区別する）。ここで**入札最小単位**を ε とし、ある正整数 I に対して $\varepsilon = \frac{\delta}{I}$ を仮定する。また $Z_\varepsilon = \{\varepsilon, 2\varepsilon, 3\varepsilon, \dots\}$ と定義する。なお以下では入札を並べたベクトル $((B_1, b_1), (B_2, b_2), \dots, (B_n, b_n))$ を (B, b) と成分の順番を変えて表示する。

ステップ 2 競売人は全参加者からの入札をもとにして、売却額が最大になるような財の割り当てを（計算機で）計算する。これは以下のような $BAP(B, b)$ (Bundle Assignment Problem) と呼ばれる整数計画問題で表される。

$$\begin{aligned} BAP(B, b): \text{maximize} \quad & \sum_{i \in N} b_i x_i = bx \\ \text{subject to} \quad & \sum_{i: B_i \ni j} x_i \leq 1 \quad (\forall j \in M), \\ & x_i \in \{0, 1\} \quad (\forall i \in N), \end{aligned}$$

なおここで $x = (x_1, x_2, \dots, x_n)^\top$ である。

ステップ 3 最大利益となる割当が唯一の時は、この割当にしたがって財が配分される。最大利益となる割当が 2 つ以上あるときは、これらのうちの 1 つの割当

をランダムに選び、その割当にしたがって財が配分される。結果として財が配分された参加者を**落札者**と呼ぶ。落札者は入札額を競売人に渡す。

上記の問題 BAP は, winner determination problem, set packing problem, weighted stable set problem, weighted clique problem, weighted independent set problem 等, 様々な名前で呼ばれる。この問題は NP-困難と呼ばれるクラスに属し, 理論的には, 解くことが困難と予想されている。しかしながら, 組合せ最適化の分野では古くから研究されており, 現在では商業用整数計画法ソフトウェアで数千変数の問題も実用的な時間で解くことができる [1]。

参加者 i の期待効用を議論するために, $\text{BAP}(\mathbf{B}, \mathbf{b})$ の最適解の集合を $\Omega(\mathbf{B}, \mathbf{b})$ とし, 参加者を以下の 3 種類に分ける。

$$\begin{aligned} P(\mathbf{B}, \mathbf{b}) &= \{i \in N \mid x_i = 1 \ \forall \mathbf{x} \in \Omega(\mathbf{B}, \mathbf{b})\}, \\ R(\mathbf{B}, \mathbf{b}) &= \{i \in N \mid x_i = 0 \ \forall \mathbf{x} \in \Omega(\mathbf{B}, \mathbf{b})\}, \\ Q(\mathbf{B}, \mathbf{b}) &= N \setminus (P(\mathbf{B}, \mathbf{b}) \cup R(\mathbf{B}, \mathbf{b})), \end{aligned}$$

ここで $P(\mathbf{B}, \mathbf{b}), Q(\mathbf{B}, \mathbf{b}), R(\mathbf{B}, \mathbf{b})$ に含まれる参加者はそれぞれ **passed**, **questionable**, **rejected** である, と呼ばれる。passed な参加者は必ず落札者として財が割り当てられ, rejected な参加者には財は割り当てられない。questionable な参加者は最適解が複数あるため, くじびきの結果として財が割り当てられる時と割り当てられない時がある。

参加者 i の期待効用は以下のように定義される:

$$U_i(\mathbf{B}, \mathbf{b}) = \begin{cases} V_i(B_i) - b_i & (i \in P(\mathbf{B}, \mathbf{b})), \\ (V_i(B_i) - b_i)\theta & (i \in Q(\mathbf{B}, \mathbf{b})), \\ 0 & (i \in R(\mathbf{B}, \mathbf{b})), \end{cases}$$

ここで $\theta = |\{x \in \Omega(\mathbf{B}, \mathbf{b}) \mid x_i = 1\}| / |\Omega(\mathbf{B}, \mathbf{b})|$ は参加者 i が落札者として選ばれる確率を表す。

4 ゲーム理論からの分析結果

4.1 効率的な資源配分とナッシュ均衡

経済学等におけるオークション研究の目的の 1 つは, 参加者の戦略的な行動によってそのオークションの結果がどのようなものになるかを調べる事である。特に重要なのは, 戦略的な行動の結果としてもオークションが効率的な資源配分を達成するかどうかで

ある。

この場合における効率的な資源配分とは, 競売人と参加者の総余剰が最大になることを示す。ここでは単純化のために売り手の財に対する価値は 0 であり, 買い手は財を購入するために十分なお金を持っており, 売り手の効用は財から得た効用とお金との合計で決まるとする。財の配分 $(S_1^*, \dots, S_n^*)$ が効率的 (efficient) な財の配分であるとは, 任意の財の配分 $(S_1, \dots, S_n)$ に対して

$$\sum_{i \in N} V_i(S_i^*) \geq \sum_{i \in N} V_i(S_i)$$

が成立することである。なおここで $(S_1, \dots, S_n)$ が財の配分であるとは, $(S_1, \dots, S_n)$ が財の集合 M の分割となっていること, すなわち $\cup_{i \in N} S_i = M$, かつ任意の $\{i, j\} \subseteq N$ に対し $[i \neq j \rightarrow S_i \cap S_j = \emptyset]$ が成立することを言う。

必要十分財オークションにおいて, もし参加者が正直に自分の真の必要十分財とその価値を申告した場合, 競売人の利益を最大にする財の割当は効率的な資源配分となる。しかし参加者は自分の真の必要十分財とその価値を正直に申告するとは限らない。そこで, ナッシュ均衡を参加者の戦略的な行動の結果と考え, ナッシュ均衡がどのようなものであり, ナッシュ均衡における財の割当が効率的な資源配分となっているかを議論する。

入札 $(\mathbf{B}^*, \mathbf{b}^*)$ が**ナッシュ均衡**であるとは, 任意の参加者 $i \in N$ の任意の入札 $(B_i, b_i) \in 2^M \times Z_\varepsilon$ に対して, $U_i(\mathbf{B}^*, \mathbf{b}^*) \geq U_i((B_i, \mathbf{B}_{-i}^*), (b_i, \mathbf{b}_{-i}^*))$ が成立することである。ここで以下のような入札額ベクトルの集合 $\mathcal{F}_\varepsilon(\mathbf{B}, \mathbf{v})$ を考える。

$$\mathcal{F}_\varepsilon(\mathbf{B}, \mathbf{v}) = \left\{ \mathbf{b} \in Z_\varepsilon^N \mid \begin{cases} b_i = v_i \ (\forall i \in R \cup Q), \\ b_i \leq v_i - k\varepsilon \ (\forall i \in P), \\ \Omega(\mathbf{B}, \mathbf{b}) = \Omega(\mathbf{B}, \mathbf{v}) \end{cases} \right\}$$

ここで k は $k \geq |\Omega(\mathbf{B}, \mathbf{v})|$ を満たす正の整数である。また, $P = P(\mathbf{B}, \mathbf{v}), Q = Q(\mathbf{B}, \mathbf{v}), R = R(\mathbf{B}, \mathbf{v})$ である。 $\mathcal{F}_\varepsilon(\mathbf{B}, \mathbf{v})$ は $\text{BAP}(\mathbf{B}, \mathbf{v})$ と $\text{BAP}(\mathbf{B}, \mathbf{b})$ の最適解の集合が一致するような入札額ベクトルの集合である。

与えられた入札額ベクトルの集合 $X \subseteq Z_\varepsilon^N$ に対して, $\mathbf{b} \in X$ が極小入札額ベクトルであるとは, 任意の $\mathbf{b}' \in Z_\varepsilon^N$ に対して, $[\mathbf{b}' \leq \mathbf{b} \text{ かつ } \mathbf{b}' \neq \mathbf{b}]$ ならば

$b' \notin X$ が成立することを言う。

定理 1 $\mathcal{F}_\varepsilon(\mathbf{T}, \mathbf{v})$ が非空ならば, $\mathcal{F}_\varepsilon(\mathbf{T}, \mathbf{v})$ の任意の極小入札額ベクトル $\mathbf{b}^*$ に対して, $(\mathbf{T}, \mathbf{b}^*)$ はナッシュ均衡となる。

$\mathcal{F}_\varepsilon(\mathbf{T}, \mathbf{v})$ においては, passed な参加者は, 自分の真の価値より低い入札額を申告するが, どのくらい低い入札額を申告しなければならないか (すなわち k の値) は, 参加者の真の価値に対する最適解の個数 $|\Omega(\mathbf{B}, \mathbf{v})|$, すなわち「同点となる落札結果の数」によって異なる。

しかし最適解の個数は常に 2^n 以下であるから, 参加者の真の価値に対する最適解の個数に関わらず以下のような結果が成立する。

系 1 以下の集合の任意の極小ベクトル $\mathbf{b}^*$ に対して, $(\mathbf{T}, \mathbf{b}^*)$ はナッシュ均衡となる:

$$\left\{ \mathbf{b} \in Z_\varepsilon^N \left| \begin{array}{ll} b_i = v_i & (\forall i \in R \cup Q), \\ b_i \leq v_i - 2^n \varepsilon & (\forall i \in P), \\ \Omega(\mathbf{B}, \mathbf{b}) = \Omega(\mathbf{B}, \mathbf{v}) \end{array} \right. \right\}$$

もし参加者の真の価値に対する最適解が唯一 ($|\Omega(\mathbf{B}, \mathbf{v})| = 1$) ならば, Z_ε^N は以下のように表せる。

系 2 もし参加者の真の価値と必要十分財に対する最適解が唯一ならば, 以下の集合の任意の極小ベクトル $\mathbf{b}^*$ に対して, $(\mathbf{T}, \mathbf{b}^*)$ はナッシュ均衡となる:

$$\left\{ \mathbf{b} \in Z_\varepsilon^N \left| \begin{array}{ll} b_i = v_i & (\forall i \in R \cup Q), \\ b_i \leq v_i - \varepsilon & (\forall i \in P), \\ \Omega(\mathbf{B}, \mathbf{b}) = \Omega(\mathbf{B}, \mathbf{v}) \end{array} \right. \right\}$$

$\mathcal{F}_\varepsilon(\mathbf{T}, \mathbf{v})$ が非空ならば, Nash 均衡は存在するが, 空であれば存在しない。しかしながら, 以下の定理により入札単位が十分に小さいならば, 必ずその Nash 均衡が存在する。

定理 2 もし ε が十分に小さい正の数ならば, $\mathcal{F}_\varepsilon(\mathbf{T}, \mathbf{v})$ は非空である。

4.2 ワルラス均衡の非存在

このようなオークションを用いずとも, 個々の財を独立したオークションで売ることによって効率的な資源配分は達成されないのであろうか。Bikhchandani (1999) はこのような財の組合せに代替性・補完性が

ある場合のオークションを考察し, (非分割財の) ワルラス均衡が存在する場合は独立したオークションで効率的な資源配分が達成されることを示した。本研究での必要十分財が存在する状況は, ワルラス均衡が存在しない場合を含んでおり各財の独立したオークションでは効率的な資源配分は達成できない。

5 組合せ最適化理論からの結果

5.1 consecutive one property

財が何らかの順序で直線上に配置され, すべての必要十分財が直線上の 1 つの区間で表される時に, これらの必要十分財の組は consecutive one property を持つという。すなわち必要十分財の組 $(T_1, \dots, T_n)$ が consecutive one property を持つとは, 任意の $i \in N$ に対して, $T_i = \{j \in M \mid l_i \leq j \leq h_i\}$ となるような (l_i, h_i) が存在することを言う (実際には財の番号を置換した, ある財の並び替えに対して成立すればよい)。

このような consecutive one property は, 物流や周波数帯のオークションにおける選好の特徴を良く表している。例えば周波数帯のオークションでは, 隣り合うような周波数帯を取得すると, その組合せ効果により 2 つの周波数帯の価値を足し合せたものよりも高い価値になると言われている。しかも各参加者が価値を持つ周波数帯は, その参加者の事業地域の特性に依存して, それぞれに異なる。Kirishna and Rosenthal [5] や Rosenthal and Wang [9] は, 直線上に配置された周波数帯を考え, “local” bidder と呼ばれる参加者は 1 つの周波数帯のみに, “global” bidder と呼ばれる参加者は 1 つの周波数帯とその右隣の周波数帯のみに選好を持ち, 特に “global” bidder は, 該当する 2 つの周波数帯に強い補完性を持つような状況を議論している。

今ここで, 参加者が連続したいいくつかの区間の周波数帯を必要十分財と考えているとしよう。このような周波数帯のオークションは consecutive one property を満たしている。

このような consecutive one property を満たす必要十分財オークションについて, 筆者達は以下のような結果を導いた。

結果 1

必要十分財の組が *consecutive one property* を満たす場合には, 必要十分財のオークションにおける最適割当問題 *BAP* は, 重みづけ有向グラフの最長路問題に変換することができる. この問題は古典的な動的計画法を用いて多項式時間で解く事が出来るので, 結果として *BAP* を多項式時間で解くアルゴリズムが存在する.

入札に対して最適な財の割り当て方法が複数あるときは, 参加者の公平を保つため, すべての最適解の中から等確率で 1 つの割当方法を選ぶ. しかしながら, これには一般にはすべての最適解を列挙しなければならず, 多くの計算時間がかかる事が多い.

結果 2 必要十分財の組が *consecutive one property* を満たす場合には, 入札に対して最適な財の割り当て方法が複数あるときに等確率で 1 つの割当方法を選ぶ多項式時間で解くアルゴリズムが存在する.

結果 3 必要十分財の組が *consecutive one property* を満たす場合には, 定理 1 で示したナッシュ均衡を, 明示的な線形不等式系の解で Z_e^N に入っているものの極小解として表現することができる.

6 まとめ

以上本論文では, 必要十分財のオークションを定義し, 戦略的な行動によっても効率的な資源配分を達成できること, 周波数オークションなどに応用できるような *consecutive one property* と呼ばれる性質を満たすならば, 競売人の利益を最大にするような財の最適な割り当てを求める, 多項式時間アルゴリズムが存在し, 更に競売人の利益を最大にするような最適解が複数あるような場合 (同点となる参加者群が複数ある

場合) に, 等確率で 1 つの解を選び出すことができるような多項式時間アルゴリズムが存在することなどを示した.

本論文では結果における証明は省いている. 興味のある方は筆者たちの研究 Matsui and Watanabe [6][7] を参考にしていただければ幸いである. 筆者らの結果は, <http://www.misojiro.t.u-tokyo.ac.jp/tomomi/TRs/> からダウンロードすることができる.

参考文献

- [1] Andersson, A., Tenhunen, M., and Ygge, F. (2000), "Integer programming for combinatorial auction winner determination," *Proc. of the Fourth International Conference on Multiagent Systems (ICMAS-00)*.
- [2] Bikhchandani, S. (1999), "Auctions of heterogeneous objects," *Games and Economic Behavior*, vol. 26, 193–220.
- [3] Bikhchandani, S. and Mamer, J. W. (1997), "Competitive equilibrium in an exchange economy with indivisibilities," *Journal of Economic Theory*, vol. 74, 385–413.
- [4] de Vries, S. and Vohra, R. (2000), "Combinatorial auctions; a survey," *Kellog School of Management, technical report*.
- [5] Krishna, V. and Rosenthal, R. W. (1996), "Simultaneous auctions with synergies," *Games and Economic Behavior*, vol. 17, 1–31.
- [6] Matsui, T. and Watanabe, T. (2001), "Sealed bid multi-object auctions with necessary bundles and its application to spectrums auctions," in *Intelligent Agents: Specification, Modeling, and Applications*, LNAI, 78–92, Springer.
- [7] Matsui, T. and Watanabe, T. (2001), "Multi-object auctions with necessary bundles," *Discussion Paper*.
- [8] Vickley, W. (1961), "Counter speculation, auctions and competitive sealed tenders," *The Journal of Finance*, vol. 16, 8–37.
- [9] Rosenthal, R. W. and Wang R. (1996), "Simultaneous auctions with synergies and common values," *Games and economic Behavior*, vol. 17, 32–55.

セキュアマルチエージェント動的計画法の設計 と組合せオークションへの適用

鈴木 幸太郎, 横尾 真

This paper presents a secure dynamic programming protocol that utilizes homomorphic encryption. By using this method, multiple agents can solve a combinatorial optimization problem among them without leaking their private information to other agents. More specifically, in this method, multiple servers cooperatively perform dynamic programming procedures for solving a combinatorial optimization problem by using the private information sent from agents as inputs. Although the servers can compute the optimal solution correctly, the inputs are kept secret even from the servers. Furthermore, we discuss the application of this protocol to various types of combinatorial auctions, i.e., multi-unit auctions, linear-good auctions, and general combinatorial auctions.

1 Introduction

In multi-agent systems, multiple autonomous agents sometimes need to solve a combinatorial optimization problem by using their private information. For example, in a combinatorial auction

where multiple goods are auctioned simultaneously, agents need to find a combination of bids for disjoint sets of goods, so that the sum of the bidding prices is maximized. This problem is called the winner determination problem and has recently become a very active research field [5][13][14][15].

If there exists a fully trusted agent, e.g., the participants can trust the auctioneer, it is possible to gather all private information relevant to the combinatorial optimization problem at this trusted agent; thus this agent can solve the problem using any available centralized optimization technique.

However, we cannot take it for granted that there exists such a trusted agent. For example, in a standard first-price sealed-bid auction [12], where the highest bidder wins and pays his/her own price, the auctioneer might collude with a particular participant and reveal information about incoming bids to that participant during the auction.

If we use a strategy-proof mechanism [12][24], such as a second-price sealed bid (Vickrey) auction, where the highest bidder wins and pays the second highest price, the information of the other participants' bids becomes useless; thus we can discourage collusion between the auctioneer and bidders. However, in a second-price sealed-bid auction, the auctioneer can increase his/her revenue by fabricating a fake bid whose price is very close to the highest bid.

Secure Multi-agent Dynamic Programming: Design and Application to Combinatorial Auctions

Koutarou Suzuki, 日本電信電話株式会社, NTT 情報流通プラットフォーム研究所, NTT Information Sharing Platform Laboratories, koutarou@isl.ntt.co.jp

Makoto Yokoo, 日本電信電話株式会社, NTT コミュニケーション科学基礎研究所, NTT Communication Science Laboratories, yokoo@cslab.kecl.ntt.co.jp

We can utilize various cryptographic technologies so that while accepting incoming bids, the auctioneer cannot learn bidding prices. For example, bidders first submit encrypted bids, and then give the auctioneer the decryption keys after the bids are closed. However, the auctioneer can utilize the information of bids for future auctions. For example, the auctioneer can learn the behavior/preference of a certain participant from the past auctions and commit similar fraud based on this information, or the auctioneer might reveal/sell such private information to others. Varian [22] described this problem as follows: *“Even if current information can be safeguarded, record of past behavior can be extremely valuable, since historical data can be used to estimate the willingness to pay. What should be the appropriated technological and social safeguards to deal with this problem?”*.

This paper aims to provide a solution to this problem by utilizing indistinguishable, homomorphic public key encryption scheme [3]. More specifically, multiple servers cooperatively solve a combinatorial optimization problem by using the private information sent from agents as inputs. Although the servers can compute the optimal solution correctly, the inputs are kept secret even from the servers. For example, in a combinatorial auction, multiple auction servers can solve the winner determination problem, i.e., they can find the combination of bids so that the sum of the bidding prices is maximized. However, the information of bids that are not part of the optimal solution is kept secret from the auction servers. More specifically, we develop a method for securely executing dynamic programming procedures [1], which are very effective and widely applied to various combinatorial optimization problems.

The application of the proposed secure dynamic programming protocol is not limited to combinatorial auctions. In various applications, it is quite

possible that multiple agents are willing to solve a combinatorial optimization problems cooperatively, but they are still concerned with privacy. For example, if a multiple network carriers cooperatively provide a particular service, they might need to find the most cost effective way for providing the service by combining their resources. However, it is quite natural that these carriers are not willing to reveal all their private information to each other. By using our secure dynamic programming protocol, these carriers can find the optimal solution without revealing their private information.

The rest of this paper is organized as follows. In Section 2, we briefly describe the overview of dynamic programming techniques. In Section 3, we detail our newly developed secure dynamic programming protocol. Furthermore, in Section 4, we describe a way of applying the proposed method to various types of combinatorial auctions, i.e., multi-unit auctions, linear auctions, and general combinatorial auctions. In Section 5, we discuss its relation to existing techniques.

2 Dynamic Programming

Dynamic programming [1] was developed by R. Bellman during the late 1950's. Dynamic programming is a powerful method that can be applied to various combinatorial optimization problems.

In the following, we use the problem of finding the longest path in the one-dimensional directed graph described in Figure 1 to illustrate the concept of dynamic programming.

This graph consists of nodes $0, 1, 2, \dots, m$ with directed links among them. A link is represented as (j, k) , where $j < k$. For each link (j, k) , the weight of the link $w(j, k)$ is defined. The goal is to find the longest path from initial node 0 to terminal node m , i.e., to find a path from 0 to m so that the sum of the weights of links are maximized. For simplicity, we assume for each node j (where $0 \leq j < m$),

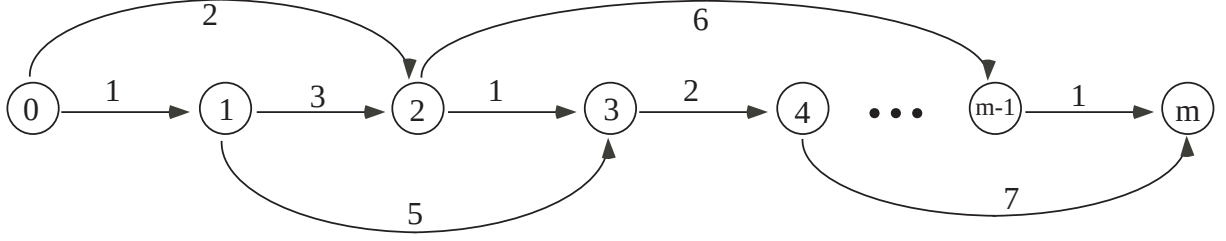


図 1 Example of one-dimensional directed graph (I)

there exists at least one link that starts from j , i.e., there is no dead-end node except m .

One notable characteristic of this problem is as follows. Assume P is the longest path from 0 to m . Then, for any node j which is on P , the last half of P , i.e., the part of P from j to m , is also a longest path from j to m . This characteristic is called the *principle of optimality*. This feature enables us to find the optimal solution of the original problem from the optimal solutions of sub-problems.

More specifically, we can obtain the length of the longest path from 0 to m by solving the following recurrence formula from node $m - 1$ to 0. In this formula, $f(j)$ represents the length of the longest path from j to m . We call $f(j)$ an *evaluation value* of node j . For terminal node m , $f(m)$ is defined as 0. For initial node 0, $f(0)$ represents the optimal solution, i.e., the length of the longest path from 0 to m .

$$f(j) = \max_{(j,k)} \{w(j,k) + f(k)\}$$

When calculating this formula, for each node j , we record the link (j,k) that gives the evaluation value $f(j)$, i.e., the link that gives $\max_{(j,k)} \{w(j,k) + f(k)\}$. We can construct the longest path by following these recorded links from 0 to m . Generalizing this technique to other types of graphs, e.g., two-dimensional multistage networks (which are used for multi-unit auctions described in Section 4), or general directed acyclic graphs, is rather straightforward.

In the rest of this paper, we describe our se-

cure dynamic programming protocol based on this path-finding problem in a one-dimensional directed graph. As discussed in Section 4, generalizing the proposed protocol to general cases is also straightforward.

3 Secure Dynamic Programming

In this section, we present our secure dynamic programming protocol and discuss its security and efficiency.

3.1 Preliminaries

Before describing our protocol, we describe a basic tool for our implementation, i.e., an *indistinguishable, homomorphic, and randomizable* public key encryption scheme. In the rest of this paper, we use ElGamal encryption [3], which has all of these properties, for describing our protocol. However, our protocol can be implemented using other encryption methods (e.g., Paillier encryption [10]) that also have these properties.

- **Public key encryption:** In public key encryption, the key used for encryption is public, so anybody can create ciphertext $E(M)$ from plain text M . On the other hand, the key used for decryption is kept secret and only the one who has the secret key can obtain M from $E(M)$.
- **ElGamal encryption:** ElGamal encryption is one instance of public key encryption. Let $q, p = 2q + 1$ be primes, and $G = \langle g \rangle \subset \mathbf{Z}_p^*$ be a cyclic group of order q generated by g .

The secret key is $x \in \mathbf{Z}_q$ and the corresponding public key is $g, y = g^x$.

Anyone can encrypt message $M \in \mathbf{Z}_p$ just using public key $g, y = g^x$, i.e., choose random number $r \in \mathbf{Z}_q$ and create ElGamal ciphertext $E(M) = (A = g^r, B = y^r M)$.

One who knows secret key $x \in \mathbf{Z}_q$ can decrypt ciphertext $E(M) = (A = g^r, B = y^r M)$, i.e., compute $B/A^x = M$.

Let us show a very simple example.

Public keys: $p = 23, g = 5, y = 2$

Secret key: $x = 2$

Please note that this setting is intended only to show a simple, easy to follow explanation; p is too small and obtaining secret key x from the public keys is easy. In standard settings, p is chosen from about 768 to 1024-bit primes.

For encrypting plaintext 5, if we choose random value $r = 5$, we obtain ciphertext $E(5) = (g^5, y^5 M) = (5^5 \bmod 23, 2^5 \cdot 5 \bmod 23) = (20, 22)$. For decrypting $E(5) = (20, 22)$, we can obtain $22/20^2 \bmod 23 = 22/9 \bmod 23 = 5$.

- Indistinguishable encryption: In ElGamal encryption, $E(M)$ is created using random number r . Thus, if the same plaintext is encrypted twice using different random numbers, these two ciphertexts look totally different and we cannot know whether the original plaintexts are the same or not without decrypting them.

For example, if we choose random value $r = 6$ for encrypting plaintext 5, we obtain ciphertext $E(5) = (g^6, y^6 M) = (5^6 \bmod 23, 2^6 \cdot 5 \bmod 23) = (8, 21)$. This ciphertext looks totally different from another ciphertext $E(5) = (20, 22)$.

Formally, if we assume that the Decision Diffie-Hellman (DDH) problem [18] is infeasible, ElGamal encryption E is indistinguishable encryption. More specifically, from $(M_0, M_1, E(M_b))$ where $b \in \{0, 1\}$, one cannot find $b \in \{0, 1\}$ with probability greater than

one half. This means ciphertext $E(M)$ leaks no information about plaintext M .

- Homomorphic encryption: Encryption E is homomorphic if $E(M_1)E(M_2) = E(M_1M_2)$ holds. If we define the product of ciphertexts $E(M_1) = (A_1, B_1)$ and $E(M_2) = (A_2, B_2)$ by $E(M_1)E(M_2) = (A_1A_2, B_1B_2)$, ElGamal encryption E is homomorphic encryption. By this property, we can take the product of two plaintexts by taking the product of two ciphertexts without decrypting them.

For example, given $E(5) = (20, 22)$ and $E(5) = (8, 21)$, we can obtain $E(5 \cdot 5 \bmod 23 = 2) = (20 \cdot 8 \bmod 23, 22 \cdot 21 \bmod 23) = (22, 2)$. Note that this operation can be done without decrypting these ciphertexts. By decrypting $(22, 2)$, we can obtain $2/22^2 \bmod 23 = 2/1 = 2$.

- Randomization: In ElGamal encryption, one can create a new randomized ciphertext $E(M) = (Ag^s, By^s)$ with random value s from the original ciphertext $E(M) = (A = g^r, B = y^r M)$. This is equivalent to make a product of $E(1) = (g^s, y^s)$ and $E(M)$. If we assume that the DDH problem is infeasible, one cannot decide whether a ciphertext is a randomized ciphertext of the original ciphertext or not.

For example, given $E(5) = (20, 22)$, and $E(1) = (9, 12)$, where $r = 10$, the resulting product is $E(5)E(1) = E(5) = (20 \cdot 9 \bmod 23, 22 \cdot 12 \bmod 23) = (19, 11)$. $E(5) = (20, 22)$ and $E(5) = (19, 11)$ look totally different so we cannot know whether the original plaintexts are the same or not without decrypting them.

3.2 Overview

The basic idea of our secure dynamic programming protocol is as follows:

- Multiple servers (called evaluators), each of which corresponds to one node of a graph, cooperatively execute dynamic programming.

- Each evaluator of node i knows only its evaluation value $f(i)$ and does not know any weight of any link.

To realize this secure dynamic programming protocol, we have to answer the following question: how can we determine the maximum of weights, and add a constant to a weight without revealing the weights themselves? The decision on how to represent and encrypt the weight is crucial to making these tasks feasible.

We are going to explain our representation that makes these tasks feasible.

- Vector representation: We represent weight w ($1 \leq w \leq n$) by encrypted weight $\vec{e}(w)$ that is the following vector of ciphertexts:

$$\begin{aligned}\vec{e}(w) &= (e_1, \dots, e_n) \\ &= (\underbrace{E(z), \dots, E(z)}_w, \underbrace{E(1), \dots, E(1)}_{n-w}),\end{aligned}$$

where $E(1)$ and $E(z)$ denote the encryption of 1 and common public element $z (\neq 1)$, respectively. n is chosen so that it is large enough to represent the length of the longest path. z is chosen so that $z^k \bmod p \neq 1$ for $0 < k < q$.

For example, we can represent weight $w = 1$ by encrypted weight $\vec{e}(w)$ using $z = 5$, $n = 4$:

$$\begin{aligned}\vec{e}(w) &= (E(z), E(1), E(1), E(1)) \\ &= ((5, 10), (2, 4), (10, 8), (4, 16)),\end{aligned}$$

if we choose r for each element as 1, 2, 3, 4, respectively. Since each encryption is done independently, i.e., we use different r for encrypting each element, they are indistinguishable from each other; we cannot tell w without decrypting each element.

- Add a constant: We can add a constant f to encrypted weight $\vec{e}(w) = (e_1, \dots, e_n)$ without decrypting $\vec{e}(w)$ nor learning w . By shifting and randomizing $\vec{e}(w)$, we can obtain

$$\vec{e'}(w + f) = (\underbrace{E(z), \dots, E(z)}_f, e'_1, \dots, e'_{n-f})$$

where e'_j is a randomization of ciphertext e_j . Due to randomization, one can obtain no information about constant f from $\vec{e}(w)$ and $\vec{e'}(w + f)$.

For example, we can add a constant $f = 1$ to $\vec{e}(w) = ((5, 10), (2, 4), (10, 8), (4, 16))$ by shifting $\vec{e}(w)$ to right by 1. Using $E(z) = (11, 7)$, where $r = 9$, we obtain $((11, 7), (5, 10), (2, 4), (10, 8))$. Furthermore, we mask the 2-nd, 3-rd, and 4-th elements by taking products with $E(1)$, i.e., $(9, 12), (22, 1), (18, 2)$, where $r = 10, 11, 12$, respectively. Finally, we obtain:

$$\begin{aligned}\vec{e'}(w + f) &= (E(z), E(z)E(1), E(1)E(1), E(1)E(1)) \\ &= ((11, 7), (5, 10) \cdot (9, 12), (2, 4) \cdot (22, 1), \\ &\quad (10, 8) \cdot (18, 2)) \\ &= ((11, 7), (22, 5), (21, 4), (19, 16)) \\ &= (E(z), E(z), E(1), E(1))\end{aligned}$$

Please note that we can perform these operations without decrypting $\vec{e}(w)$. Also, if we compare $\vec{e}(w)$ and $\vec{e'}(w + f)$, we cannot know the amount of the shift without decrypting them.

- Find the maximum: We can find the maximum of encrypted weights $\vec{e}(w_i) = (e_{1,i}, \dots, e_{n,i})$ without leaking information about the weights that are not the maximum as follows. Consider the componentwise product of all weights

$$\prod_i \vec{e}(w_i) = (\prod_i e_{1,i}, \dots, \prod_i e_{n,i}).$$

Observe that, due to the homomorphic property, the j -th component of this vector has the following form

$$c_j = \prod_i e_{j,i} = E(z^{S(j)})$$

where $S(j) = \#\{i \mid j \leq w_i\}$ is the number of weights that are equal or greater than j . Notice that $S(j)$ monotonically reduces as j increases. To find the maximum of these weights, we decrypt c_j and check whether decryption $D(c_j)$

is equal to 1 or not from $j = n$ to $j = 1$ until we find the largest j s.t. $D(c_j) \neq 1$. This j is equal to $\max_i \{w_i\}$, i.e., the maximum of the weights.

For example, we can represent another weight $v = 3$ by encrypted weight

$$\begin{aligned}\vec{e}(v) &= (E(z), E(z), E(z), E(1)) \\ &= ((20, 22), (8, 21), (17, 19), (16, 3)),\end{aligned}$$

if we choose r for each element as 5, 6, 7, 8, respectively.

To find $\max\{w + f, v\}$, we create the product of $\vec{e}'(w + f)$ and $\vec{e}(v)$, i.e.,

$$\begin{aligned}\vec{e}'(w + f) \cdot \vec{e}(v) &= (E(z)E(z), E(z)E(z), E(1)E(z), E(1)E(1)) \\ &= ((11, 7) \cdot (20, 22), (22, 5) \cdot (8, 21), \\ &\quad (21, 4) \cdot (17, 19), (19, 16) \cdot (16, 3)) \\ &= ((13, 16), (15, 13), (12, 7), (5, 2)) \\ &= (E(z^2), E(z^2), E(z), E(1)).\end{aligned}$$

By decrypting the 4-th element, we obtain $B/A^x = 2/5^2 \bmod 23 = 2/2 \bmod 23 = 1$. By decrypting the 3-rd element, we obtain $B/A^x = 7/12^2 \bmod 23 = 7/6 \bmod 23 = 5 = z$. Thus we can be convinced that $\max\{w + f, v\}$ is equal to 3.

Using these processes, we can find the maximum of weights, and add a constant to a weight; thus we can perform dynamic programming procedures using this vector representation.

3.3 Protocol

Now, we will present details of our protocol. There is a weight publisher $P_{(j,k)}$ for each link (j, k) , and an evaluator T_j for each node j . In an auction setting, a weight publisher corresponds to a bidder, and an evaluator corresponds to a part of the multiple auction servers.

The basic idea of this protocol is as follows. The weight publisher $P_{(j,k)}$ encrypts its weight $w(j, k)$ using T_j 's encryption function. Evaluator T_k (who cannot decrypt this information) then calculates

$w(j, k) + f(k)$. Next, evaluator T_j calculates $f(j)$ using this information without knowing $w(j, k)$.

The details of the protocol can be described as follows. For a while, we assume that each evaluator acts honestly, i.e., it does not try to decrypt the information it does not need to know to execute the protocol. We show how to prevent the evaluator from learning the information it does not need to know by introducing multiple evaluators for each node in Section 3.5.

- *Preparation* : Evaluator T_j of node j generates a secret key and a public key of encryption function E_j of node j .

- *Publish weight* : Weight publisher $P_{(j,k)}$ of link (j, k) decides its weight $w(j, k)$ ($1 \leq w(j, k) \leq n$) and publishes encrypted weight:

$$\vec{e}_{(j,k)}(w(j, k)) = (\underbrace{E_j(z), \dots, E_j(z)}_{w(j,k)}, \underbrace{E_j(1), \dots, E_j(1)}_{n-w(j,k)})$$

of its weight $w(j, k)$ using encryption function E_j of node j .

- *Find optimal value* : Evaluator T_j of node j determines $f(j) = \max_{(j,k)} \{w(j, k) + f(k)\}$ from $\vec{e}'_{(j,k)}(w(j, k) + f(k))$ using the method described in Section 3.2. Evaluator T_j then creates $\vec{e}'_{(i,j)}(w(i, j) + f(j))$ from $\vec{e}_{(i,j)}(w(i, j))$ using the method described in Section 3.2, and publishes it. By iterating these procedures, we can obtain optimal value $f(0)$, i.e., the length of the longest path.

Please note that since T_j has the secret key of encryption E_j , it can decrypt $\vec{e}_{(j,k)}(w(j, k))$ or $w(j, k)$. We discuss how to prevent T_j from illegally learning $w(j, k)$ in Section 3.5.

- *Find optimal path* : After finding the optimal value, we trace the nodes back to determine the optimal path. Assume that evaluator T_{i_0} announces that node j_0 attains maximum $f(i_0)$. Evaluator T_{j_0} who knows maximum $f(j_0)$ then finds node k_0 that attains maximum $f(j_0)$, i.e., $w(j_0, k_0) + f(k_0) = f(j_0) = \max_{(j_0,k)} \{w(j_0, k) + f(k)\}$,

by decrypting the $f(j_0)$ -th component of $\vec{e}_{(j_0,k)}(w(j_0,k) + f(k))$ for all k and checking that the decryption of the component is equal to 1 or z (if it is equal to z , the node in the one that attains the maximum). Evaluator T_{j_0} then announces that node k_0 attains maximum $f(j_0)$. By iterating these procedures, we can find the optimal path.

3.4 Example

Here, we give an example of protocol execution using the one-dimensional directed graph described in Figure 2. There are four nodes $\{0, 1, 2, 3\}$, four links $\{(0, 1), (1, 2), (1, 3), (2, 3)\}$, with weights $w(0, 1) = 3, w(1, 2) = 2, w(1, 3) = 1, w(2, 3) = 0$.

First, weight publishers $P_{(0,1)}, P_{(1,2)}, P_{(1,3)}, P_{(2,3)}$ publish encrypted weights (where $n = 6$).

$$\begin{aligned} \vec{e}_{(0,1)}(w(0,1)) &= (E_0(z), E_0(z), E_0(z), E_0(1), E_0(1), E_0(1)), \\ \vec{e}_{(1,2)}(w(1,2)) &= (E_1(z), E_1(z), E_1(1), E_1(1), E_1(1), E_1(1)), \\ \vec{e}_{(1,3)}(w(1,3)) &= (E_1(z), E_1(1), E_1(1), E_1(1), E_1(1), E_1(1)), \\ \vec{e}_{(2,3)}(w(2,3)) &= (E_2(1), E_2(1), E_2(1), E_2(1), E_2(1), E_2(1)), \end{aligned}$$

respectively.

By decrypting $\vec{e}_{(2,3)}(w(2,3))$, evaluator T_2 knows $f(2)$ is 0, and creates

$$\begin{aligned} \vec{e}_{(1,2)}(w(1,2) + f(2)) &= \\ (E_1(z), E_1(z), E_1(1), E_1(1), E_1(1), E_1(1)) \end{aligned}$$

by randomizing $\vec{e}_{(1,2)}(w(1,2))$.

To find $\max\{w(1,2) + f(2), w(1,3)\}$, evaluator T_1 creates the product of $\vec{e}_{(1,2)}(w(1,2) + f(2))$ and $\vec{e}_{(1,3)}(w(1,3))$, i.e.,

$$\begin{aligned} \vec{e}_{(1,2)}(w(1,2) + f(2)) \cdot \vec{e}_{(1,3)}(w(1,3)) &= \\ (E_1(z^2), E_1(z), E_1(1), E_1(1), E_1(1), E_1(1)), \end{aligned}$$

and decrypts the 6-th, 5-th, 4-th, 3-rd, and 2-nd components and finds z . Thus T_1 is convinced that $f(1) = \max\{w(1,2) + f(2), w(1,3)\}$ is 2. T_1 then

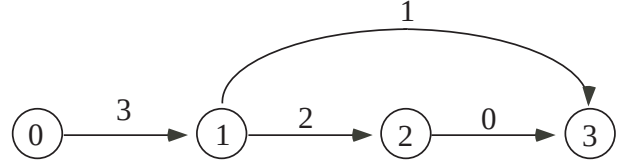


図 2 Example of one-dimensional directed graph (II)

shifts $\vec{e}_{(0,1)}(w(0,1))$ by $f(1)$, randomizes it to yield:

$$\begin{aligned} \vec{e}_{(0,1)}(w(0,1) + f(1)) &= \\ (E_0(z), E_0(z), E_0(z), E_0(z), E_0(z), E_0(1)) \end{aligned}$$

that represents

$$f(0) = w(0,1) + \max\{w(1,2) + f(2), w(1,3)\} = 5.$$

To find the optimal path, evaluator T_1 decrypts the $f(1)$ -th (=2-nd) component of $\vec{e}_{(1,2)}(w(1,2) + f(2))$ and $\vec{e}_{(1,3)}(w(1,3))$ and obtains z and 1, respectively. Thus T_1 is convinced that link $(1,2)$ attains $\max\{w(1,2) + f(2), w(1,3)\} = 2$.

3.5 Security

In this section, we discuss the security of our protocol. To classify the obtained degree of security, let us define the levels of information leakage as follows:

level 0: No information about weights, except the weights of the links in the optimal path, is leaked.

level 1: Besides the information leakage of level 0, one evaluation value $f(j)$, where node j is not on the optimal path is leaked.

level k : Besides the information leakage of level 0, evaluation values $f(j_1), \dots, f(j_k)$, where nodes $j_1, \dots, j_k$ are not on the optimal path, are leaked.

level m : Besides the information leakage of level 0, evaluation values $f(j)$ for all nodes $0 \leq j < m$ are leaked.

A passive adversary is defined as the one who tries to obtain secret information based on the published information of the protocol and all informa-

tion available to collusive participants. We assume that all collusive participants perform the protocol properly, i.e., they cannot actively manipulate the protocol.

From the indistinguishability of ElGamal encryption, one can obtain no information about weight w from $\vec{e}(w)$. Due to randomization, one can obtain no information about constant f from $\vec{e}(w)$ and $\vec{e}'(w+f)$. Hence, if a passive adversary cannot collude with any evaluator, our protocol guarantees level 0 information leakage, i.e., our protocol leaks no information about weights except the optimal path.

On the other hand, if the adversary can collude with evaluator T_j , the adversary can access to all information about $w(j,k)$, since T_j knows the secret key of E_j . To prevent this, we can implement the function of evaluator T_j by plural evaluators $T_j^1, T_j^2, \dots$ and introduce a threshold value t , so that when $t+1$ evaluators for node j cooperate, they can decrypt E_j , but cannot do so if there are less than or equal to t evaluators.

More specifically, each distributed evaluator $T_j^1, T_j^2, \dots$ has only a share of the secret key; thus any collusion of t (or less than t) evaluators cannot decrypt E_j [17]. In the preparation phase, the secret and public key are generated in a distributed way [11] and each distributed evaluator $T_j^1, T_j^2, \dots$ has only a share of the secret key. To find the optimal value and path, the decryption is performed in a distributed fashion by each distributed evaluator $T_j^1, T_j^2, \dots$ that has a share of the secret key, through a secure channel.

By introducing multiple evaluators for each node, our protocol guarantees that the information leakage is at most level 1 even if a passive adversary collude with one evaluator T_j^i , i.e., the adversary learns only evaluation value $f(j)$. In general, the information leakage of level 1 can be quite acceptable since the adversary cannot discover any addi-

表 1 communication complexity

	pattern	round	volume
publish weight	$P_{(j,k)} \rightarrow BB$	l	n
find optimal value	$T_j \leftrightarrow BB$	m	$n \times v$
find optimal path	$T_j \leftrightarrow BB$	m	1

tional weights of links from $f(j)$ except the optimal path. An exception occurs when node j has only one link (j,m) . In this case, the evaluation value $f(j,m)$ is equal to $w(j,m)$. If this is the case, we can prevent the leakage of this information by making weight publisher $P_{(j,m)}$ act as an evaluator for node j . Furthermore, if a passive adversary can collude with t evaluators, our protocol guarantees that the information leakage is at most level t .

If the participants can accept level m information leakage, then we can drastically reduce the required number of servers, i.e., we need to use only $t+1$ evaluators, while the original protocol requires $m(t+1)$ servers. Each evaluator T^i does the operations of T_j^i for all j in the original protocol. Even if a passive adversary colludes with t evaluators, the adversary can obtain no more information than level m .

Even if the obtained level of information leakage is m , i.e., evaluation values $f(j)$ for all j are leaked, the adversary can learn the exact value of weight $w(j,k)$ only if (j,k) is the only link directed from j . In that case, $f(j) - f(k) = w(j,k)$. Otherwise, the adversary cannot learn the exact value of any weight except those in the optimal path.

3.6 Efficiency

Table 1 shows communication pattern, round complexity, and data volume per round of each step of our protocol. Here, l is the number of links, m is the number of nodes, v is the maximum number of links for a node, and n is the number of components of an encrypted weight that must be greater than the length of the longest path. We consider that

there is bulletin board BB , and that each agent sends data to BB to publish it, and accesses to BB to find the published data.

Since the round complexity is proportional to the number of links or nodes, our scheme can handle large numbers of auction items. However, since the data volume and computational complexity is proportional to the range of weights, our scheme becomes costly when the total sum of the prices is large.

4 Application to Combinatorial Auctions

In this section, we discuss the application of our secure dynamic programming protocol to several types of combinatorial auctions, i.e., multi-unit auctions, linear-good auctions, and general combinatorial auctions.

4.1 Multi-unit Auctions

In multi-unit auctions, m units of an identical item are auctioned. Each bidder B_k ($1 \leq k \leq N$) declares his/her bidding price $b_k(j)$ for each quantity j , where $1 \leq j \leq m$. The goal is to find the allocation that maximizes the sum of the bidding prices.

The optimal allocation in a multi-unit auction can be obtained by solving the following recurrence formula [20] [21].

$$f((1, j)) = b_1(j),$$

$$f((k, s)) = \max_{0 \leq j \leq s} \{f((k-1, s-j)) + b_k(j)\}.$$

In this formula, $f((k, s))$ represents the value of the optimal solution of a sub-problem, i.e., optimally allocating s units among k participants, i.e., bidders $B_1, B_2, \dots, B_k$. The optimal solution is given by $f((N, m))$.

Applying our secure dynamic programming protocol to this problem is rather straightforward. We use plural evaluators for each node (k, s) . Further-

more, for node $(1, j)$, its evaluation value $f((1, j))$ represents the bidding price of Bidder B_1 for j units. To prevent the leakage of this information, bidder B_1 should act as an evaluator for node $(1, j)$.

4.2 Linear-Good Auctions

In a linear-good auction [13] [20], there exist m goods $G = \{1, 2, \dots, m\}$. These goods are sequentially ordered. Each bidder B_k ($1 \leq k \leq N$) bids his/her bidding price $b_k([l_k, u_k])$ for an interval $[l_k, u_k] \subseteq G$ of goods, i.e., a bidder wants to obtain a continuous sequence of goods. We assume each bidder bids for a single interval (or equivalently bids for several intervals independently). The result of the auction is the allocation of the m goods that maximizes the sum of all bidders' bidding prices.

Auctions for linear goods can be used for time scheduling (e.g., for the allocation of time slots in a conference room), or for the allocation of a one-dimensional space (e.g., for parts of a seashore), or spectrum right auctions in which each bidder (wireless carrier) wants a continuous frequencies to minimize interference between carriers [8].

This problem can be directly mapped into the problem of finding the longest path in a one-dimensional graph. Consider the graph with $m+1$ nodes $\{0, 1, 2, \dots, m\}$, i.e., there exists an initial node 0 and nodes that correspond to goods. For Bidder B_k , who bids his/her bidding price $b_k([l_k, u_k])$ for interval $[l_k, u_k]$, we represent his/her bid as a link between $(l_k - 1, u_k)$ and set its weight as $b_k([l_k, u_k])$. Also, we add a dummy link between $(j, j+1)$ for each $0 \leq j < m$, whose weight is 0, to prevent a node from becoming a dead-end.

The allocation of the m goods that maximizes the sum of all bidders' bidding prices corresponds to the longest path from node 0 to node m , and can be securely computed by our secure dynamic programming protocol. Also, our method can handle multiple links between two nodes.

The idea of linear-good auctions can be generalized to route auctions, in which each bidder bids for a path in a general graph. The result of the auction is the path from the start node to the destination node that maximizes/minimizes the sum of all bidders' bidding prices.

One application of route auctions can be a transportation task assignment problem, in which the auctioneer wants to transport his/her cargo from a starting city to a destination city. There are several transportation companies. Each company bids his/her price to carry the cargo for a path (which can be only a part of the total journey), and the auctioneer chooses the combination of paths that minimizes the total cost. This problem can be formalized as finding a shortest path in a graph, and so can be solved using the secure dynamic programming protocol presented in this paper.

4.3 General Combinatorial Auctions

In a general combinatorial auction, there exist multiple different goods $G = \{1, 2, \dots, m\}$. Each bidder B_k bids his/her price $b_k(S)$ for each subset/bundle $S \subseteq G$. The goal is to find the allocation of goods that maximizes the total price.

As described in [13], this problem can be solved by dynamic programming as follows. For each bundle S , $b(S)$ represents the highest price of the bundle. For simplicity, we assume that each bundle has at least one bid (otherwise, we can put a dummy bid with price 0). For each bundle $S \subseteq G$, we create a node $(S, |S|)$, where $|S|$ represents the number of goods in S . For each node $(S, |S|)$, we place the following directed links:

- $((S, |S|), (\{\}, 0))$, where $w((S, |S|), (\{\}, 0)) = b(S)$
- $((S, |S|), (C, |C|))$, where $C \subset S$ and $|C| \geq |S|/2$, and $w((S, |S|), (C, |C|)) = b(S \setminus C)$.

We show an example of nodes and links of a combinatorial auction, where $G = \{1, 2, 3\}$ in Figure 3.

The optimal allocation is given by the longest path from the initial node (G, m) to the terminal node $(\{\}, 0)$. It is clear that this problem can be solved using the secure dynamic programming protocol presented in this paper. Our method can handle multiple links between two nodes. Therefore, we don't need to pre-select the bid with the highest price for each bundle.

One obvious disadvantage of this approach is that the number of nodes becomes very large, i.e., 2^m . However, this seems somewhat inevitable if we are to solve this problem using dynamic programming, since the winner determination problem of a general combinatorial auction is NP-complete [13].

5 Discussions

There have been various works on secure auction servers and protocols [4][7][9]. However, as far as the authors know, there has been no other research on secure dynamic programming protocols nor on secure combinatorial auction servers/protocols based on dynamic programming techniques.

Kikuchi [7] developed a secure $M+1$ -st price auction protocol, in which multiple units of an identical item are auctioned, but each participant is assumed to require only one unit. Suzuki and Abe [19] also developed a secure $M+1$ -st auction protocol based on Homomorphic Encryption. Our method can be applied to the cases where each participant wants to buy multiple units.

It is well known that any combinatorial circuit can be computed securely using general-purpose multi-party protocols [2][6]. Therefore, if we can construct a combinatorial circuit that implements a dynamic programming algorithm, in principle, such an algorithm can be executed securely (thus we don't need to develop a specialized secure protocol for dynamic programming). However, to execute such a general-purpose multi-party protocol, for each computation of an AND gate in the circuit,

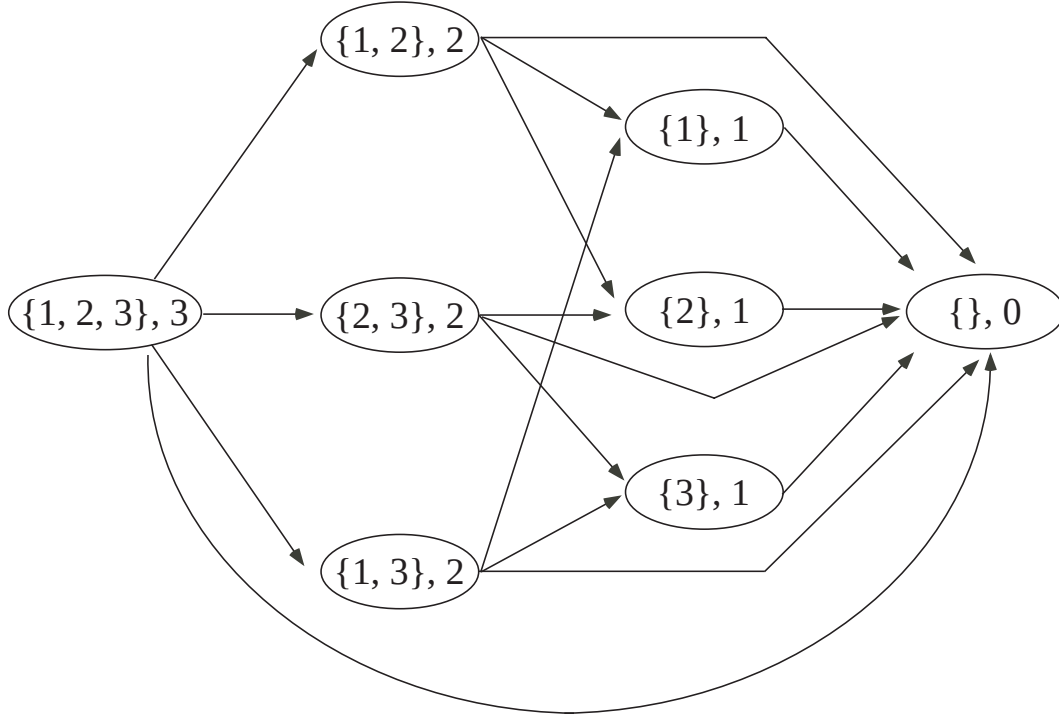


図 3 Example of graph for combinatorial auction

the evaluators must communicate with each other. Using such a general purpose multi-party protocol for a large-scale dynamic programming application is not practical at all due to the required communication costs.

Naor et al. [9] proposed a general method for executing any auction protocol including combinatorial auction protocols based on a technique called the garbled circuit [23]. This method does not require interactive communications among multiple evaluators. However, to apply this method to a dynamic programming application, we first need to construct a combinatorial circuit that implements a dynamic programming algorithm, then scramble this circuit so that an agent executing this circuit cannot learn the actual contents of the circuit. Using this method for large-scale dynamic programming applications, including combinatorial auctions, would be rather difficult.

6 Conclusion

In this paper, we presented a secure dynamic programming protocol that utilizes homomorphic encryption. By using this method, multiple servers/evaluators cooperatively perform dynamic programming procedures for solving a combinatorial optimization problem by using the private information sent from agents as inputs. Although the evaluators can compute the optimal solution correctly, the inputs are kept secret even from the servers. Furthermore, we discussed its application to several combinatorial auctions, i.e., multi-unit auctions, linear-good auctions, and general combinatorial auctions.

Although dynamic programming is a very powerful, widely-used combinatorial optimization method, applying dynamic programming techniques to general combinatorial auctions is not fea-

sible for large-scale problems, since it requires an exponential number of nodes. Currently, we are investigating the protocols for securely computing average-case efficient algorithms [5][15][16].

参考文献

- [1] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of 20th ACM Symposium on the Theory of Computing*, pages 1–10, 1988.
- [3] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. on Information Theory*, IT-31(4):469–472, 1985.
- [4] M. K. Franklin and M. K. Reiter. The design and implementation of a secure auction server. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1986.
- [5] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computation complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.
- [6] O. Goldreich, S. Micli, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of 19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.
- [7] H. Kikuchi. (M+1)-st-Price auction protocol. In *Proceedings of Financial Cryptography 2001*, 2001.
- [8] T. Matsui and T. Watanabe. Sealed bid multi-object auctions with necessary bundles and its application to spectrum auctions. In *Proceedings of the 4th Pacific Rim International Workshop on Multi-agents (PRIMA-2001)*, pages 78–92. Springer-Verlag, 2001. Lecture Notes in Artificial Intelligence 2132.
- [9] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, 1999.
- [10] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Proceedings of EUROCRYPT '99*, pages 223–238, 1999.
- [11] T. Pedersen. A threshold cryptosystem without a trusted party. *Proceedings of EUROCRYPT '91*, pages 522–526, 1991. Lecture Notes in Computer Science 547.
- [12] E. Rasmusen. *Games and Information*. Blackwell, 1994.
- [13] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [14] Y. Sakurai, M. Yokoo, and K. Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 30–37, 2000.
- [15] T. Sandholm. An algorithm for optimal winner determination in combinatorial auction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
- [16] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A fast combinatorial algorithm for optimal combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pages 1102–1108, 2001.
- [17] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [18] M. Stadler. Publicly verifiable secret sharing. *Proceedings of EUROCRYPT '96*, pages 190–199, 1996. Lecture Notes in Computer Science 1070.
- [19] K. Suzuki and M. Abe. M+1-st price auction using homomorphic encryption. Technical Report ISEC 2001-11, IEICE, 2001.
- [20] M. Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 98–103, 2000.
- [21] S. van Hoesel and R. Müller. Optimization in electronic markets: examples in combinatorial auctions. *Netnomics*, 3:23–33, 2001.
- [22] H. R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First Usenix Workshop on Electronic Commerce*, 1995.
- [23] A. C. Yao. How to generate and exchange secrets. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.
- [24] M. Yokoo, Y. Sakurai, and S. Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167–181, 2001.