サーベイ: 分散人工知能小問題集

大沢 英一 (ソニーコンピュータサイエンス研究)
沼岡 千里 (ソニーコンピュータサイエンス研究所)
石田 亨 (NTT コミュニケーション科学研究所)

概 要

This paper addresses some canonical problems in distributed artificial intelligence (DAI), specifically in the domain of cooperative problem solving, and modeling rational/autonomous agents. These canonical problems, Tileworld, Pursuit problem, and Prisoner's dilemma, are examined in a way such that the major research issues in distributed artificial intelligence are delineated. By analyzing current methodologies and approaches, we hope to identify core problems, and discuss future direction for DAI research.

1 はじめに

分散人工知能 (以下，DAI と略す場合がある)，マルチエージェント領域における協調問題解決，そして合理的/自律型 エージェントのモデル化などに関するいくつかの標準的問題 (Tileworld, Pursuit problem, Prisoner's dilemma) を取り上げることにより当該研究領域における問題点と目的を整理し，またこれまでに提案されてきた解法/アプローチを比較することによりそれらの諸性質と有効性を考察する．

一般的な観点からみると，分散人工知能とは，比較的小規模な多数の知的システム (エージェントと呼ばれる) が相互作用 (協調，競合) することにより構成する (分散) システムを理解/解析/記述すること，そしてそのようなシステムを構築し統合することが目的であると言える．しかしながら分散人工知能研究の歴史を振り返ってみると，それは必ずしも何らかの統一的な基礎理論から出発して発展してきたものではなく，むしろ，具体的な応用分野 (例えば音声理解であるとか移動体の監視など) において問題を発見し，それらをモデル化し，そして理論や解法を与えてきたという傾向がある．よって，これまでに分散人工知能研究が達成してきたことを検討しようとすると，先ず (1) 当該研究領域における問題点が何なのかということに関して十分に一般化された問題領域がなかなか見当たらない，また (2) そこで開発されてきた解法/方法論の有効性/一般性をはかることが困難である，そして (3) 追試ができないために議論がかみあわず，様々なモデルが提案されるが評価することが不可能になっている，等の問題に直面する．これらの反省から，誰もが理解でき，追試できる小問題が必要であるという認識が生まれつつある．

本稿では，分散人工知能研究において標準的であると思われる具体的ないくつかの問題 (Tileworld [Pollack and Ringuette 90], Pursuit problem [Benda *et al.* 85, Stephens and Merx 89], Prisoner's dilemma [Davis 77]) を取り上げることにより，当該研究領域における中心課題，目的などを整理し，またこれまでに提案されてきた解法/方法論を考察することにより，それらの諸性質と有効性について議論し，また今後の研究の方向性を探ることを試みる．誤解を避けるために最初に述べておくと，本稿は分散 AI/マルチエージェントシステム/自律エージェントなどに関する広範なサーベイではない．むしろ，特定の問題をいくつか取り上げることにより，これまで

の研究成果から得られたいくつかの解法の性質と有効性を具体的に理解するためのものである[1].

本論文の各章では，先ず上記各問題を紹介しそれらの主性質について述べ，つづいて各問題に深く関係するいくつかの関連研究について述べ，章の最後でそれらの関連研究の関係をまとめた後に各問題に関して考察する．以下では，各章の内容をもう少し具体的に述べると同時に，各問題を選択した背景に関して簡単に述べる．

第 2 章では資源制限を受けたエージェントのアーキテクチャを評価するための動的実験環境 Tileworld を導入し，動的環境の特性に対するエージェントの**即応** (reactivity) と**熟考** (deliberation) のトレードオフ，そしてメタコントロールの役割などに関して述べる．複数のエージェントが独立に動作するような環境は，個々のエージェントにとっては動的な環境にほかならないが，Tileworld ではマルチエージェントシステムのような動的で予測不能な環境で動作する**資源制限を受けたエージェント**の適切なアーキテクチャに関して評価する一例となっている．

第 3 章では Pursuit problem (追跡問題) により，単一の大域目標 (global goal) を持つ複数エージェントの集団において，各エージェントが個々の局所的なプランと大域目標達成のためのプランをどのようなメカニズム/戦略により整合させるかという問題を考える．この問題では，大域目標を達成するために複数のエージェントが組織を作る（もしくは作らない）場合，(1) どのような組織形態が可能で，(2) 各組織形態にはどのような性質を持っていて，また (3) 各組織の問題解決の効率はどうか，などを詳細にみる．この問題を通して組織的問題解決のいくつかの性質について理解することが目的である．

第 4 章では Prisoner's dilemma (囚人のジレンマ) と呼ばれる非零和非協力ゲーム（繰り返し行なわれる場合も含む）を通して，利害が完全には対立しない状況での協調的行動選択の問題について考える．このゲームでは，各プレーヤは追跡問題などとは異なり同一の目的を共有しているわけではない．このような状況で様々な戦略が考えられる中において，先ずいくつかの戦略の対戦からそれらの持つ性質を見る．次に様々な戦略の混在する集団からの協調関係の発現する可能性について考察する．そして協調関係を持つうえでの合理性の役割に関して説明する．囚人のジレンマとこれに関連する研究をしらべ ると，このゲームが当てはまるようなマルチエージェントシステムのある状況において，各エージェントがどのような戦略をとることが望ましのかに関する理解を与えてくれる．

2 Tileworld – 動的環境でのエージェント・アーキテクチャ –

2.1 問題の定義

Tileworld はチェスボードのような 2 次元格子上に，エージェント，タイル，障害物及び穴を配置した，計算機上の仮想的な実験環境である．エージェントは 1 つのセル上にちょうど収まる大きさで，上下左右に単位時間当たり 1 セル動くことができる．タイルもセル上に位置し，エージェントに押されることによって，隣接するセルへと滑べるように移動する．障害物はいくつかのセル上に位置し，移動することはない．穴もいくつかのセルから成り，タイルによって埋めることができる．即ち，穴と同じセル上にタイルが運ばれると，穴とタイルは消滅し空白のセルに戻る．穴を構成する全てのセルが埋められると，エージェントに得点が与えられる．得点は穴毎に予め決められており，エージェントもそれぞれの穴の価値を知らされている．Tileworld でのエージェントの目標は，穴を埋めることによって可能な限り多くの得点をあげることである．

[1] 分散人工知能の基本的研究課題にはこれら以外にも，エージェント間の信念の不整合の問題やエージェント間通信における common grounding の問題などがある．これらに関する広範なサーベイとしては [Bond and Gasser 88] などを参照されたい．

Tileworldのシミュレーションは以下のように行なわれる．穴，タイル，障害物はシミュレーションに先だって設定されたパラメータに従って現れたり，消えたりする． エージェントは，変動する環境の中を動き回り，タイルを運び穴を埋めることが要求される．このように常に変動する動的環境を対象とするところが，Tileworld と従来の人工知能の題材，例えば Blocks World などとの違いである．Tileworldで動作するエージェントは限られた時間でプランを構築し，判断を下さなければならない．Tileworldは，このような資源的に制約された (resource bounded) エージェントのアーキテクチャを評価する環境として，提案され利用され始めている．

2.2 問題の背景及び目的

人工知能における (狭義の) 問題解決とは，状態空間の中で初期状態から出発し，状態遷移を生じさせる動作の実行を繰り返すことによって，目標状態に至ることである．古典的なプラニングは，目標状態に至る動作系列を，実行前に求めるもので，ＮＰ困難な問題であることが知られている．したがって，動的な世界に即応しなければならない場合には，古典的プラニングとは異なる，より瞬間的に反応し行動するプラニング手法が考案されなければならない．この種の研究は，実時間プラニングとも反応的 (reactive) プラニングとも呼ばれ，Tileworldが発表される以前から活発に討論されてきた．[2]

例えば，動的環境におけるエージェントのアーキテクチャを論じた研究としては，[Georgeff and Lansky 87] が SRI で開発した PRS（Procedural Reasoning System）が知られている．PRSは，手順的なプランを記述するためのツールであるが，通常の手続き型プログラミング言語とは異なり，環境の変化に迅速に反応することを特徴としている．PRS インタプリタは，その時点で登録されている目標（goal）と実世界の状態の双方を参照して，起動条件が満足される手続き (Knowledge Area と呼ばれる) を選び出し実行する．これによって，PRSは目標駆動（goal-driven）であるだけでなく，実世界の状態の変化を常に反映した判断を行なうことができる．

また， [Agre and Chapman 87] は routine と呼ばれるパターン化された動作を単位として，人間の日常の行動をモデル化しようとした．Pengo ゲームにおけるペンギンの反応的な行動を興味深い方法で説明している.「ペンギンは蜂から遠ざかろうと可能な限り速く走る．途中で障害物にぶつかるとそれを蹴飛ばす．さらに走って壁に到達する.」この一連の行動を手続き的に書くと次のような２重ループとなる.「障害物にぶつかるまで走る．ぶつかれば障害物を蹴飛ばす．これを繰り返す.」しかし，Agreらは人間がこのような２重ループを実行しているとは考えない．むしろこの動作は，(1) 蜂が来れば逃げる，(2) 障害物にぶつかると蹴飛ばすという２つの routineから構成されると考える．２重ループは，単にこれらの routineが状況に応じて選択された結果だと考える．また，routine の選択は知覚情報を入力とする組合せ回路によって行われるとしている．

これらの動的環境に反応 (reactive) するエージェントアーキテクチャの研究は，新しい見地からの研究を推進するものであったが，一方では問題解決の最適性は勿論のこと，果たして問題を本当に解決できるのかどうかさえ疑問視された．古典的な人工知能研究が得意とする信頼性の高い，しかし時間のかかるプラニング技術と，反応的な行動とのトレードオフに興味が集まったのは当然の帰結である．Wilkinsはロボットの制御には，戦略的なプランナと反応的なプランナの双方が必要であると主張している．戦略的なプランナはタスクレベルのプラニング（例えば書斎で本を捜す手順の構成）を担当し，反応的なプランナはより低レベルのプラニング（例えば障害物を避けながら書斎に到達する）を担当する [Wilkins 88].

Tileworldは，まさしくこのような研究の流れを背景に提案されている．古典的プラニングと反応

[2] この節は，石田 亨，“知識表現と動的世界 – 最近のプランニング研究から –” 人工知能学会誌， Vol. 5， No. 2， pp. 146-153， 1990，からの引用を含んでいる．

的行動のトレードオフを定量的に評価し，エージェントアーキテクチャの提案に役立てようとするものである．以下では，[Bratman *et al.* 88]，[Pollack and Ringuette 90]，[Kinny and Georgeff 91] に発表された Tileworld を用いた研究内容を紹介する．なお，Tileworld を用いたものではないが，[Durfee and Lesser 88] が，同様のトレードオフを複数エージェントによる協調問題解決で論じているので興味のある読者は参照されたい．

2.3 研究事例: 熟考と即応のトレードオフ

IRMA(the Intelligent Resource-Bounded Machine Architecture) は動的な環境におけるエージェントのアーキテクチャを目標としたもので，Bratman ら [Bratman *et al.* 88] により提案され，Pollack ら [Pollack and Ringuette 90] により Tileworld 上で評価された．IRMA では，環境の変動に即応するために，エージェントが実行中のプランは再考されたり，必要があれば放棄されることもありうる．しかし一方， 環境の変動への過剰な反応を抑えるために，エージェントのプランは妥当なレベルで安定していなければならないとしている．IRMA では，反応的 (reactiveness) であることの対極の概念として熟考 (deliberation) が用いられる．但し熟考は，従来から人工知能で議論されてきた means-end reasoning を行なうプランナを指すわけではなく，むしろ意思決定理論で議論されてきた意思決定に相当するものである．より詳しくは，means-end reasoning は与えられた目標をいかに (how) 達成するかを決定するもので，熟考は多くの可能性の中から何を (what) 追求すべきかを求めるプロセスとしている． 一方で，Means-end reasoning の過程で多くの選択候補 (option) が洗い出されれば，それらは熟考の対象となるとしている．それにもかかわらず，熟考を means-ends reasoner の一部と考えない理由は，解の品質や所要時間が異なる複数の reasoner を熟考の入力とすることを許すためである．

IRMA の大きな特徴は，熟考の対象となる選択候補を制限するためのフィルタリングメカニズム (filtering mechanism) を備えていることである．フィルタリングメカニズムは，(1) 新たに生成された選択候補とその時点で実行されているプランが両立 (compatible) するか，あるいは (2) 現在実行中のプランのどの部分を抑止しどの部分を重要視すれば双方の重ね合わせ (override) が可能となるかを調べる．このフィルタを通過した選択候補だけを熟考の対象とすることによって，妥当なレベルでの安定性を保証しようとする．

この研究で用いた Tileworld の実験パラメータ (knobs と呼ばれている) を以下に示す．

- *dynamism:* 穴の出現率
- *hostility:* 障害物の出現率
- *variability of utility:* 穴の得点のばらつき
- *variability of difficulty:* 穴の大きさとタイルからの距離のばらつき
- *hard/soft bounds:* 穴が時間と共に得点を下げるか (soft bounds)，あるいははっきりした時間切れがあるか (hard bounds) の相違
- *act/think rate:* 動作と思考の時間配分
- *threshold level:* フィルタのしきい値
- *sophistication of the deliberation mechanism:* 熟考プロセスの洗練度

Tileworldではこれらのパラメータを変更することによって，様々な角度からエージェントアーキテクチャの評価が可能である．

実験では，フィルタは以下のように構成された．(1) 両立性検査 (compatibility checking) はごく単純なものが用いられた．新たな選択候補は，“どこどこの穴を今すぐ (あるいは後ほど) 埋める” という形で生成される．この生成は，例えば新しい穴が発見された時に行なわれる．両立性検査では，新たに今すぐ埋めようとする穴が，現在エージェントが埋めつつある穴と異なる場合に限り，両立しないとする．両立性検査をパスした選択候補は熟考の対象となる．(2) 重ね合わせ検査は，両立しない選択候補に対して行なわれる．新たに選択候補となった穴の得点と，現在埋めようとしている穴の得点を比べ，もしその差がフィルタのしきい値を越えていれば，重ね合わせ可能とし熟考の対象とする．例えばしきい値がゼロであれば，得点の大きな穴は全て熟考の対象となる．穴が新しければ埋めるまでに時間的により余裕があると考えられる時には，しきい値をマイナスに設定することもある．なお，しきい値を-∞に設定すると，全ての選択候補が熟考の対象となる．

一方，熟考プロセスは，その洗練度によって以下の 2 種が用意された．(1) 単純な方法では，得点のより高い穴が選択される．本来この程度では熟考とは言えないが，評価のために考えられたものであろう．(2) より複雑な方法は以下の式で表される労力当たりの得点 (Likely Value: LV) のより高い穴を選択するというものである．

$$LV(h) = \frac{score(h)}{dist(a,h) + \sum_{i=1}^{n} 2 \times dist(h,t_i)}$$

ここで，$score(h)$ は穴の得点，$dist(a,h)$ はエージェントと穴の距離，n は穴を埋めるのに必要なタイルの数，$dist(h,t_i)$ は穴から i 番目に近いタイルまでの距離である．

環境が変動 (穴や障害物，タイルなどの出現消滅) する速度を変えて初期的な実験が行なわれた．環境の変動に合わせてエージェントの移動速度も速くなる．しかし考える速度は変化しないので，相対的に考えるコストが大きくなるよう設計された．また，人間がこのゲームを行なった場合も併せて評価されている．評価結果として，(1) 環境の変動速度が遅い間はフィルタのしきい値は低くて良い (即ち，多くの選択候補を熟考の対象とする) が，速度が上がるとフィルタのしきい値を上げていく (即ち，熟考の対象を絞り込む) 方が効率が良い，(2) 熟考プロセスとして LV を計算する手法 (即ち，熟考プロセスの洗練度が高い方) が単純な手法 (熟考プロセスの洗練度が低い方) に比べ常に効率が良い，(しかし，より長時間を要する熟考の手法を導入すると，必ずしも長時間考える方が良いということにはならないと思われる) (3) 人間は環境の変動が速くなると急速に得点が減少する，等が報告されている．この評価結果は，精度においても，量においてもまだまだ不十分であり，あくまで現在進めている実験の初期的な結果であるとされている．

2.4 研究事例: Commitment の効果

[Kinny and Georgeff 91] は Tileworld を用いて commitment の効果を定量的に測定しようと試みた．ここで，彼らが用いる commitment と言う用語の意味を説明しておこう．Commitment の定義は [Cohen and Levesque 90] によって与えられている．そこでは，commitment は達成されるまで持続する目標 (goal) として定義されている．しかしこの研究では，プランに対する commitment を考え，エージェントが一度決めたプランをどの程度持続して持ち続けるかを，プランに対する commitment の度合 (degree of commitment) と呼んでいる．

Kinny らは，この研究で Tileworld の環境を実験に利用しているが，Pollack らのエージェント・アーキテクチャは特に利用していない．ここでは，独自に 3 種のエージェントを導入し評価の対象としている．大胆な (bold) エージェントは決してプランを変更しない．反対に用心深い (cautious)

エージェントは常に (1 動作毎に) 再プラニングを試みる．その中間的なものとして，ノーマル (normal) なエージェントを導入する．このエージェントは 4 動作毎に再プラニングを試みる．4 という数字に特殊な意味があるわけではなく，大胆さと用心深さの中間的なものを評価に加えるのが目的と思われる．

この実験では障害物やタイルは考慮されていない．穴はエージェントが到達した時点で埋められたと見なされ消滅する．従って，もともとの Tileworld より，かなり単純化された問題を対象としている．評価の軸は，(1) 環境の変動率 (γ) と (2) エージェントの効率 (ε) である．環境の変動率が高くなるにつれて，穴が出現消滅する頻度が増大する．これは，問題解決過程で，穴が突然消えることによって目標が消失したり，近くに穴が出現することによってより良い目標が生まれたりすることを意味する．エージェントの効率は，全部の穴を埋めた場合に与えられる総得点の内，そのエージェントがどの程度を埋めることに成功したかを示す比率である．

実験はエージェントのプラニングに要する時間を様々に変化させながら行なわれた．得られた結果は以下の通りである．(1) 環境の変動がなければエージェントの効率 (ε) はプラニング時間の長短にかかわらず 1 であるが，変動速度が大きくなるにつれてこの値は減少し，プラニング時間の長短にかかわらず 0 に収束する．(2) プラニング時間が長い場合には，大胆なエージェントが用心深いエージェントより，環境の変動速度に関わらず効率が良い．(3) プラニング時間が短くなるにつれて，特に環境の変動がある程度以上大きい領域で変化が起こりはじめる．まずノーマルなエージェントが大胆なエージェントより効率が良くなり，ついで用心深いエージェントも大胆なエージェントを上回るようになる．(おそらくプラニング時間が 0 になれば，あらゆる環境の変動速度に対して用心深いエージェントが最も効率が良くなるであろうから，この結果はごく自然である．)

実験ではさらに，大胆なエージェントに合理性 (rationality) を導入する．例えば，環境の変化を察知し，少なくとも穴が消滅すれば再プラニングを行なうようにする．さらに，近くに穴が生じた場合にも再プラニングを行なうようにする．このような合理性の導入によって，大胆なエージェントの効率がかなり改善されることが示されている．

2.5 まとめ

人工知能分野においてモデル化の果たす役割は大きい．しかし同時に，提案されたモデルは，いかに人工的に知能を構成するのに寄与しうるかという尺度で評価されなければならない．Tileworld の提案は，エージェントアーキテクチャの定量的評価に一歩踏み出したという点で評価されるべきだろう．Tileworld を理解しにくいものにしているのは，問題設定の複雑さである．穴の出現消滅は目標の出現消滅を意味する．障害物の出現消滅はプランが最適な状態であり続けることがないことを意味する．穴の得点は目標の優先度を表す．穴を複数のタイルによって埋められなければならないことにしているのは，目標を達成するためのコストに変化を持たせるためだろうか．この例題を研究に用いる場合には，Kinny らのように，研究の目的に合わせて，Tileworld の部分問題を切り出すべきだろう．Tileworld を用いた研究はその緒についたばかりで，紹介した 2 組の論文から一般性のある新たな知見が得られるには至っていない．常識的に結論できることを，実験で追認したに留まる．知覚できる範囲を無限大に設定する等，問題設定が単純で，解法に特別な技術を要しないことが，結果をつまらないものとしている原因であろう．ともあれ，同じ例題を用いて定量的評価を指向する複数の研究が開始されたことに意義を見い出し，今後の研究に期待したい．

3 追跡問題 (Pursuit Problem) – 協調のための組織的枠組 –

3.1 問題の定義

ここでは，[Benda *et al.* 85] の内容に従って，問題の定義を行なう．

青い四つののエージェントが赤い一つのエージェントを囲い込むことを要求される．各エージェントは無限に広がる２次元のグリッド上を動く．全ての青いエージェントと赤いエージェントによる一つの動きがサイクルをなす．

赤いエージェントは，以下の規則に従って行動する．

- 水平または垂直方向に１ステップ動くことができる．またその位置に留まることも可能である．
- 青いエージェントがいるグリッドに移動することはできない．

赤いエージェントは，もし与えられたサイクルにおいてどこにも移動できないとき，すなわち青いエージェントによって４隅を囲まれてしまったとき，とらえられてしまう．

一方各青いエージェントは，以下の規則に従って行動する．

- 水平または垂直方向に隣あったグリッドのひとつに移動できる．
- 他の青いエージェントがいる場所へ移動できる．
- 赤いエージェントのいる場所へは移動できない (赤いエージェントを取り囲むことが目的である)．[3] こういうわけで，青いエージェントは最大４つの方向に移動できる．もし移動できる位置の内で最も好ましい位置に赤いエージェントがいる場合には，移動しないことも可能である．

3.2 問題の背景及び目的

複数エージェントによる**協調作業**は，マルチエージェントシステムの特徴の一つである．ある問題を解決するために，一人で解決することが不可能な問題であっても，複数のエージェントが協力することによって達成することが可能になることがある．例えば，分散人工知能においては初期のころから関心が持たれているテーマとして，分散センシングの問題があるが，これなどは協調によってはじめて実時間処理が可能になる例の一つであろう．

複数のエージェント (あるいはサブシステム) の協調というものを考える場合，従来的にはまず閉じた全体と言うものを考える．この見方においてはまずシステム全体の中でどのような状態が存在するかを分析 (あるいは定義) し，その状態の中で調整されるべきものをチェックし，そしてエージェントのその状態に対する協調方法を制約として定義する．

マルチエージェントシステムでは全体的な問題は定義されない．また例え定義したとしても予期せぬ事象の発生によって，変化するかもしれない．この場合全体という概念はもはや意味を持

[3] この性質は次のような意味で本質的である．青いエージェント同士が同じ場所を共有した場合，見方同士であるのでどちらかが相手を殺すことはない．しかし赤いエージェントと青いエージェントが同じ場所を共有してしまうと赤は殺されてしまう．この場合，問題の目的は赤いエージェントを生かして取り囲むことなので，青も赤を殺さないように移動しなければならない．こういうわけで青いエージェントは他の青いエージェントのいる場所に移動できるが，赤いエージェントのいる場所には移動できないのである．もちろん，複数の青いエージェントが同じ場所に移動してしまえば，赤いエージェントにとっては抜け道が多く残るので，このような青の行動は赤にとって好ましい結果を生むことになる．

たない．全てのエージェントが部分情報しか持たず，一つのエージェントだけでは解決できない問題を解決する手段として協調というものを考えていかなければならない．

協調について考えるための一つのヒントはエージェント間の相互作用の形態について考えることである．人間社会においてそうであるように，エージェントは相互作用を通じて協調し合うことができる．この相互作用の形態は様々であり得るが，多くの場合通信によって行なわれる．協調においては，お互いに意見の一致を確認することが求められるわけであるが，この確認を行なう過程はエージェント間の関係によって大きく作用される．例えば，会社において相手が上司であれば，部下はその意見に従うことによって協調するかも知れないし，互いに利害を共有している場合には，資源を譲り合うことによって協調するかもしれない．

この追跡問題は，このような背景において協調的作業に対してエージェント間の関係が与える影響について調査することを目的として提案された．各エージェントの目的は同じであるが，それらは限られた能力しか持たず，世界に関する部分情報しかもちあわせていない．このような状況で，エージェント達がどのような組織形態を作り出す時，最も効果的に問題を解決できるだろうか．Benda [Benda *et al.* 85]，Stephens [Stephens and Merx 89]，及び Levy [Levy and Rosenschein 91] らの研究においては，これが基本的問いかけとなっている．

一方 Gesser らの研究 [Gasser *et al.* 89] は少し趣を異にしている．彼らの疑問はマルチエージェントシステムにおいてそもそもなぜ組織が必要とされるのかという，より哲学的なところから発している．ここでは効率はとりあえず問題とはされない．彼らは組織によってエージェントの問題解決が制約されるのではなく，エージェントの問題解決過程が組織を必要とし，その結果として組織が形成されると考える．これは開放型環境において動作するシステムにとって，非常に本質的な組織の見方である．Benda や Stephens らの negotiating agents や Levy らのゲーム理論に基づいて行動するエージェントも，この観点から組織をとらえていると考えられる．

以下ではここで述べてきた関連研究について，できる限りそれらの論文の記述に忠実に個々に解説を行なう．まずはじめに Benda らのオリジナルの研究を紹介する．引続き Stephens らによって行なわれた研究を紹介する．これら二つの研究は異なる結論を引き出しており，それらの結論については十分に考慮される必要がある．その後，Levy らが最近示したアプローチを紹介する．彼らはゲーム理論に基づくエージェントの行動に関心を持っており，この研究もその関心の一部としてとらえることができる．最後に Gasser らによる研究を紹介する．先にも述べたように，この研究は以上３つの研究とは異なる目的を持って発表されたものである．

3.3 研究事例: 知識源の間の最適協調について

ボーイング社の Benda らは [Benda *et al.* 85] において，“複数の知的エージェントが協力して一つの解を求める際に，エージェントの間に存在する組織構造がどのような影響を与えるか，“という問題に取り組んでいる．この問題を説明し評価する目的で，3.1 節で定義されたような問題を作り上げた．[4] この追跡問題に関してはその後もいくつかの研究が行なわれている．

Benda らの青いエージェントは，3.1 節で述べたもののほかに次のような行動規則を持っている．

- 視界が限られている．
- 他の青いエージェントを見ることはできない．しかしそれぞれに与えられた組織的制約の範囲内で通信することはできる．[5]

[4]後に Stephens と Merx が [Stephens and Merx 89] において，この問題のシナリオに対し“Pursuit Game”という名称を与えている．

[5]詳しいことが論文で述べられていないので憶測の域をでないが，青いエージェントは他のエージェントを見る能力

- その視界に入った場合または赤いエージェントの位置が他の青いエージェントによって伝えられた場合においてのみ，赤いエージェントを検知できる．

組織構造は，青いエージェント間に張られたリンクの種類を反映している．また協調のモードは，組織構造を直接反映したものである．青いエージェントは，データを要求したり，強制的に制御したり，動きを交渉したりすることによって協調する．

Benda らの関心は先に述べたように，エージェントの組織構造が問題解決に与える影響である．彼らにとって最も基本的な組織構造は２つのエージェントの関係によって与えられる．そこで Benda らはエージェント間の以下の３つ基本的な関係をエージェントの組織構造として定義している．

1. Communicating agents: 二つのエージェントの間にリンクが存在する．このリンクに沿って，一端のエージェントが，もう一端のエージェントに対しデータを要求することができる．全く通信できないようなエージェントも，リンクに対して強度を関連付けることによってモデル化される．これはタイプ A の組織と呼ばれる．
2. Nagotiating agents: このタイプのリンクの両端にいるエージェントは，データを要求したり，追求するゴールを交渉したりすることによって相手と通信する，無関係なエージェントである．これはタイプ B の組織と呼ばれる．Communicating agents が，相手からデータを要求する目的以外に通信を利用することはできないのに対し，negotiating agent は，データを要求したりする他に，相手と移動に関する交渉を行なうこともできる．
3. An agent controlling another one: 二つのエージェントが指向性リンクによって結合されている．このリンクの視点に位置するエージェントは，終点に位置するエージェントを完全に制御することができる．これはタイプ C の組織と呼ばれる．

これら３つのリンクを仮定することによって，もし n 個のエージェントがいた場合，$3^{n*(n-1)/2}$ 種類の組織形態が考えられる．例えば４つのエージェントの場合，この数は 729 になる．しかしながら，この内のほとんどは不自然な組織形態[6]である．Benda らは，自然な形態の組織の中から９つの組織を選び，それらの組織形態に関して評価を行なっている．これら９つの組織形態は，既に述べた二つのエージェントからなる組織構造，タイプ A，B，及び C を基本構造と考えた場合，以下の置き換え規則の適用によって，ノードを置き換えることによって派生される組織構造である．

S(X,Y) を，タイプ X の組織構造のどちらかのエージェントをタイプ Y の組織構造によって置き換えるような関数を示すとしよう．この S(X,Y) によって，Y の組織の両端のエージェントと，X の置き換えられなかった側のエージェントとの関係が，以下のように決定される．

1. Y がタイプ A または B の場合: Y の両端のエージェントと X の中の置き換えられなかったエージェントとの関係は，X のエージェント間の組織関係を継承する．
2. X がタイプ B であり，Y がタイプ B または C である場合: Y がタイプ B であれば，これは１の特殊なケースである．Y がタイプ C であれば，B の置き換えられなかったノードと C の始点がタイプ B の組織関係で結ばれる．
3. X がタイプ C であり，その終点がタイプ A または C の Y によって置き換えられる場合: Y がタイプ A であれば，これは１の特殊なケースである．Y がタイプ C であれば，三つのエージェントが階層的に結合された組織構造となる．

実際 Benda らが定義した組織構造は以下の９つである．

を持っていないと思われる．従って他の青いエージェントの位置を知るためには，通信によって位置を確認する以外に方法はない．

[6]例えば，１つ目が２つ目を制御し，２つ目が３つ目を制御し，３つ目が４つ目を制御し，さらに４つ目が１つ目を制御している場合などは，この不自然な例に入るであろう．

表 1: 性能評価の比較 (50 回の平均値)

Organization A	Moves B	Transactions Cost C	Control Cost D	Costs per Move E = (C+D)/B
a	9.6 (2.9)	473 (137)	0 (0)	49.1
b	9.3 (3.6)	387 (141)	11 (4)	42.8
c	9.6 (4.5)	293 (130)	21 (9)	33.7
d	8.8 (3.0)	280 (93)	19 (6)	33.9
e	8.4 (3.3)	320 (118)	18 (7)	40.2
f	6.7 (3.4)	72 (33)	23 (10)	14.2
g	7.1 (3.2)	98 (40)	24 (10)	17.2
h	7.4 (2.9)	127 (46)	25 (9)	20.5
i	7.7 (3.3)	186 (73)	26 (10)	27.5

a S(S(B,B),B): 全てのエージェントが互いに交渉し合うような関係にある場合.

b S(S(B,B),C): 三つのエージェントが互いに交渉し合う関係にあり，そのうちの一つがもう一つのエージェントを制御する場合.

c S(S(B,C),C): 二つのエージェントが互いに交渉し合う関係にあり，これらのおのおのが一つずつエージェントを制御する場合.

d S(S(B,C),A): 二つのエージェントが互いに交渉し合う関係にあり，そのうちの一つが二つの通信するエージェントを個々に制御する場合.

e S(S(B,C),C): 二つのエージェントが互いに交渉し合う関係にあり，そのうちの一つがもう一つのエージェントを制御し，それがまた残る一つのエージェントを制御する場合.

f S(S(C,A),A): 一つのエージェントが，三つの互いに通信し合うエージェントを個々に制御する場合.

g S(S(C,A),C): 一つのエージェントが，二つの通信し合うエージェントを個々に制御し，通信し合うエージェントのうちの一つが残る一つを制御する場合.

h S(S(C,C),A): 一つのエージェントが，二つの通信し合うエージェントを制御するエージェントを制御する場合.

i S(S(C,C),C): ４つのエージェントが階層的な制御関係にある場合.

これら９つの組織構造に対する評価結果を表 1 に示す．表中の各エントリーは５０回の実行結果の平均値である．また括弧内の数値は，標準偏差である．Moves は赤いエージェントがとらえられるまでに要した全移動ステップである．トランザクションコストとはゲームが開始してから終了するまでに要する通信量であり，この例の場合 (タイプ A または B の組織において) コミュニケーションリンクが使用された回数である．Control Cost は (タイプ C の組織において) コントロールリンクが使用された回数である．

この結果から彼らが得た結論は，以下のようなものである．

- タイプ fの組織構造が最も効率の良い組織構造である．すなわち，最も移動コスト及びトランザクションコストが少ない．
- トランザクションコストは，組織構造の複雑さを直接反映する．コミュニケーション量は組織構造におけるボトルネックによって反映される．また交渉が組織間の協調における主なオーバーヘッドとなっている．
- タイプ fから iまでによって代表される単一の組織構造 [7]の方が，トランザクションコストの面で優れている．

簡潔に述べると，“システム全体として階層的な制御系統を持つ組織の方が，問題解決において優れている”ということである．彼らはその理由として階層的組織では通信量を制限できることをあげている．この知見に関しては，次の Stephens らの論文で反証が述べられている．

3.4 研究事例: DAI システムの性能に影響するエージェントの組織

Stephens らは“単一の問題に関して複数エージェントが協力する分散問題解決システム”を目的として，追跡問題を用いた分析を行なっている [Stephens and Merx 89]．ここで与えられている問題の枠組は，エージェントをグリッド上でなく，四角の領域に配置し，移動できる領域を 30×30 に制限しているといった実装上の違いを除けば，Benda らのものとほとんど同じである．

Stephens らは青いエージェントの行動規則として 3.1 節で述べたものの他に次のようなものを仮定する．

- 青いエージェントは自分の現在位置とエージェントの捕獲位置として選ばれた位置までの直線距離に基づいて動きを決める．[8] この距離の計算は，A^* グラフ探索におけるヒューリスティック関数 h(n) と同一のものである．
- 捕獲位置に到達すると，青いエージェントは動かなくなる．

Benda らと同様に，エージェント間の関係が問題解決に与える影響を比較するために，４つの異なる制御・通信の政策をとる青いエージェントの組織を考える．これらは，autonomous-agent，limited-communication，negotiating-agent，及び controlling-agent である．これらの各々に対して，以下のように制御戦略が設定される．[9]

制御戦略　これらは，cooperation，organization 及び dynamics に関して規定されている．cooperation は相手となんらかの協力を行なうかどうかを示す尺度である．情報交換も協力の一種である．よって cooperation において中立的であるとは，一切の情報交換あるいは通信によって獲得された情報に基づく行動決定を行なわないことを意味する．organization は Benda らのいうようなリンクでつながれた組織構造が存在するかどうかを示す尺度である．よって，組織構造において自由であるということは，一切そのようなリンクが存在しないことを述べている．最後に dynamics は，organization で述べられた組織構造においてエージェント間のリンクの関係が時間を経て変化するのか，変化するとすればどのように変化するのかを示す尺度である．

Autonomous-Agent System

[7]つまり，４つのエージェントが一つの階層的組織に所属する場合．

[8]ここで青いエージェントは視界に制限を与えられていない．すなわち，いつでも赤いエージェントの位置を検知することができる．この事実により，エージェントは赤いエージェントの位置に関して通信する必要はなくなる．

[9]論文では通信戦略も記述されているが，特にあとの評価に関係ないのでここでは省略する．

Cooperation エージェントは他のエージェントに対し全く中立的である．[10] 彼らは位置をセンサによって赤いエージェントのいる位置を知る以外，情報を得る手段を持たない．

Organization Benda らのいうような組織的構造存在しない．彼らはそのような構造的制約から全く自由である．

Dynamics エージェント間の組織的な関係は静的であり，決して変化しない．つまり，始めから終りまで組織的構造が作られることはない．

Limited-Communication System これは Benda らのいう communicating agents と同じであると考えてよい．

Cooperation 青いエージェントの一つが捕獲位置に入ったならば，他の青いエージェントはそこに移動することができない，という規則を設ける．この場合，ある捕獲位置を占めているエージェントはそれを他のエージェントに譲ることはないので，完全に中立ではない．

Organization 通信リンクがエージェント間に存在する．捕獲位置を決める場合，他のエージェントがある捕獲位置を占めていることがわかったなら，計画を変更する．つまり先に捕獲位置を占めたものが組織的に他に優先する．

Dynamics 捕獲位置を占める順序によって階層構造が作られていく．[11]

Negotiating-Agent System これは Benda らのいう negotiating agents と同じであると考えてよい．

Cooperation 全ての青いエージェントは同じ目的に対して協調的であり，意図や計画を通信を通じて共有することができる．

Organization ある種の組織構造 (同盟・非同盟) が交渉を通じて作られる．

Dynamics 彼らの位置選択は全ての動きに対して決定され，機会主導的にエージェント間の関係が変化する．

Controlling-Agent System これは Benda らのいう an agent controlling another one と同じであると考えてよい．

Cooperation 従属的なエージェントは，制御エージェントに完全に協力する．つまり制御エージェントの要求通り行動する．

Organization 制御エージェントと従属的なエージェントの間にマスタ- スレーブ関係が存在する．つまり，従属的なエージェントは完全に制御エージェントによって制御される．エージェント間の関係は静的である．

Dynamics 制御関係は決して変化しない．

---

[10] つまり相手の要求に従うことはない．これは通信手段がないのであるから自明である．また，赤いエージェントがいる場所以外ならどこにでも移動できる．例えば 2 つ以上の青いエージェントが同じ場所に移動することも可能である．よって，位置を譲り合うために協調し合う必要性もない．

[11] 一つのエージェントが最初に捕獲位置に入ったとしよう．すると他のエージェントはこの捕獲位置に入らないことに関して，最初のエージェントによる間接的な制御を受けることになる．このようにして場所を介して間接的な制御を行なう階層関係が作られていく．

性能評価　各制御・通信方法の性能は，移動コスト，ヒューリスティック効率，及び青いエージェントによって作られる重心の位置 (capture, escape, stalemate) に基づいて判断される．ヒューリスティック効率は，E = 1/N である．ここで N は正規化ヒューリスティックコストであり，

$$N = \frac{T}{\sum_{j=1}^{4} D_{1j}}$$

ここで T はトータルヒューリスティックコストであり，

$$T = \sum_{i=1}^{n} \sum_{j=1}^{4} D_{ij}$$

である．また n は開始から赤いエージェントを捕獲するまでに要したステップ数であり，ヒューリスティックコスト D_{ij} とはこの場合，エージェント j の i 回目の動作において，現在位置からそのエージェントが赤いエージェントを捕獲する位置として設定した場所までの直線距離である．

ここでこれらの式の持つ意味をもう少し考えてみよう．正規化ヒューリスティックコスト N とは，各シナリオにおいて分散の仕方が違っていることを考慮に入れて，ヒューリスティックコストを正規化したものである．実際，これは全ての青いエージェントの全ての動きにおけるヒューリスティックコストの総和を全ての青いエージェントの一回目の動作におけるヒューリスティックコストで割ったものである．ではヒューリスティック効率 E とは，直感的に何を意味するのだろうか．N は赤いエージェントを捕獲するのに要した移動回数を正規化したものなので，結局 E は収束速度というものに相等するものである．というわけで，E が大きければ大きいほど，青いエージェント達が赤いエージェントの周りに早く収束することを意味する．[12]

重心の位置が赤いエージェントに近ければ，それだけ捕獲できる機会が濃厚であることを示している．しかしながら，赤いエージェントの回りの捕獲位置の内３つまでが占有されており，かつ重心が唯一占有されていない捕獲位置の方向にない場合，青いエージェントは決して赤いエージェントを完全に補間することはできない．この状況が stalemate と呼ばれる．

性能評価には６つのシナリオが使用される．ここでは，各シナリオについて参考のため，赤を基準 (0,0) とした四つの青いエージェントの位置を示すことにする．

シナリオ 1 (-5, 0), (-4, 2), (-3, 1), (-3, 3)

シナリオ 2 (-1, 6), (0, -2), (5, -1), (5, 6)

シナリオ 3 (-2, 5), (-1, -1), (4, -2), (4, 5)

シナリオ 4 (-15, -4), (-5, -1), (5, -8), (5, 2)

シナリオ 5 (0, -4), (0, 5), (4, -1), (4, 10)

シナリオ 6 (-3, 0), (-1, -5), (3, 5), (6, 1)

これらのシナリオの持つ意味は，Benda らの論文ほど明確には書かれていない．ここで記述されていることは，これらのシナリオが以下に示す二つの難しさを提示するようなものに分類されるということである．

[12] 表 2 の (ヒューリスティック効率を表示している)Efficiency の欄 を見た読者は少し妙に思われることであろう．なぜなら E の定義から，0 < E < 1 でなければいけないからである．この点に関して我々はいろいろと論じたのであるが，原論文ではこれについて何も書かれていないので，ここではこれ以上これについてコメントすることはできない．しかしながら，いずれにしても E の大きい方が，早く赤いエージェントの周辺に収束することだけは間違いないようである．

タイプ１ これはシナリオ４と５のようなものである．青いエージェントは赤いエージェントからかなりはなれている．結果として，相当量の移動を余儀なくされる．そのため，例えば２，３の青いエージェントが赤いエージェントに迫ったとしても，空いている側から赤いエージェントが逃げてしまうことが起こり得る．

タイプ２ これはシナリオ１と４のようなものである．このタイプの問題は，４つの青いエージェントとそれらの重心が，赤いエージェントの一方の側に集中し，しかも離れているような場合に起こり得る．この場合計画立案が非常に困難になる．なぜなら４つのエージェントから捕獲位置までの距離がほとんど同じになってしまうからである．シナリオ１がこのタイプの中で最も難しい初期配置を持ったものである．

６つのシナリオはこれらを比較するために選ばれたようである．

６つのシナリオに対する結果は Levy らの結果と併せて表 2 に示される．この表から Stephens らが考察したものは以下のような点である．

- シナリオ 1 において negotiating-agent システムは controlling-agent システムよりも 46% 程度ヒューリスティック効率が良いという結果が得られた．これは，controlling-agent システムの初期段階において制御エージェントが他の青いエージェントの動きを抑制してしまっているためである．
- シナリオ４では controlling-agent システムが他のものよりもヒューリスティック効率において勝っている．しかしながら一般的には，negotiating-agent システムがランダムに動き回る赤いエージェントを捕まえることにおいて，より効率的で柔軟性があることが示された．
- 全てのシナリオを通じて，negotiating-agent システムはより少ない動作数で赤いエージェントを捕らえることに成功した．
- controlling-agent システムは常に最大のステップ数を要している．一方で，このシステムだけが常に赤いエージェントを捕らえることに成功している．
- autonomous-agent と limited-communication システムは，シナリオ６においてのみ赤いエージェントを捕らえることに成功した．

彼らの結論は Benda らのものと異なっている．つまり目的が動的に変化する場合，negotiating-agent の方が controlling-agent よりも動的追従性において優れているというのである．ただし４項目目で示されているように，完全に目的を遂行するためにはやはり controlling-agent の方が優れていることもわかる．

Stephens らの考察は少ない例に対して強過ぎる言明を与えていないこともない．特に，２項目目や３項目目は早急な結論を述べているようにも感じられる．

3.5 研究事例: DAI と追跡問題に対するゲーム理論的アプローチ

Levy らは，同じ追跡問題を題材として，ゲーム理論的戦略に基づくエージェントの組織を考案している [Levy and Rosenschein 91]．さらに Stephens らの研究で用いられた６つのシナリオを利用して，この組織を用いた結果を評価している．ここではその解法について紹介する．

Levy らは論文でも述べているように，明らかに追跡問題を “ 標準問題 (canonical problem) ” と考えている．彼らがこの論文で示したいことは，彼らがこれまで行なってきたゲーム理論的方法論が，エージェントが問題を解決することにおいてどのように役立つかを示すことにある．よっ

て，Benda らの研究や Stephens らの研究が組織的構造が問題解決にどのように影響を及ぼすかに関心があったのとは異なる問題意識にたって，この追跡問題を利用している．彼らの立場において，エージェントが他のエージェントと協力するかどうかは，個々のエージェントがそれぞれの利得をどのように算出するかにかかっている．彼らの関心は，この問題のように協調してある目的を解くためにエージェントがどのような支払政策をとるべきか，またその場合問題解決においてどのような影響がでるかという点にあると思われる．

各エージェントが取る支払い政策 (payoff policy) は，以下の２点を考慮している．

1. 最終状態において，できる限り早く一局に集中すること．さらに，この支払い政策が，究極的にシステムの全体的な関心事 (すなわち捕獲) を表現していること．
2. 支払い関数は，エージェントがその行為を調整し，赤いエージェントをブロックするように仕向けるようなものであること．ここで使われる調整戦略は，以下の二つを考慮したものである．

   - 赤いエージェントから青いエージェントへの距離の変化量の合計 $\sum_{i \in S} d^i$.
   - 赤いエージェントがブロックされた脱出口の数 k_S (副収入).

   ここで協調の重要性を強調する意味で，k_S がある自然数の k_S 乗という形で協調された方が良い．この論文では，この数を５としている．[13]

解法は非協力ゲームと協力ゲームを混合したものである．まず各エージェントは，他のエージェントと明確に協調していない場合それ自身で行動意思を決定しなくてはならないので，各エージェントはまず副収入を考慮しないで４プレイヤー非協力ゲームを解かなければならない．このゲームは G と表現される．ここで

$$G = (S^1, S^2, S^3, S^4; H^1, H^2, H^3, H^4)$$

である．S^i はエージェント i の戦略であり，

$$S^i = \{South, West, North, East, Stay\}$$

である．また H^i(s) は，エージェント i の戦略 s に対する支払いを規定するための支払い関数である．協力ゲームのフェーズにおいて，全てのエージェントはある戦略に対する Shapley 値 [14]の合計を計算し，それを共有する．まとめると，エージェントは毎回協力フェーズと非協力フェーズからなる動作決定機構によって，次の動作を決定するということである．

Levy らのシステム設計の目的の一つは，できる限りエージェント間の通信を減少させることにある．実験結果から，通信が全くない状態では，一般的な解を求めることが不可能であると結論付けている．例えば，赤いエージェントを中心 (0,0) に置き，青いエージェント 1 が (0,3)，2 が (1,1)

[13] この数を設定する場合には，協調提携と非協調提携の差を作り出すように配慮されねばならない．例えば２であった場合には，その差を作り出す方法がないことが示されている．一つのエージェントが赤いエージェントの方向に１だけ移動したことを考えよう．この場合，$2 + 2^1 = 4$ となる．一方，４つのエージェントがそれぞれ赤いエージェントを全くブロックしないような方向に１づつ移動したとしよう．この場合，$4 + 0 = 4$ となる．そこで，協調的に移動しようが非協調的に移動しようが，その差を示すことができない．こういうわけで，５という数が選ばれている．

[14] これは対象性 (symmetry)，効率性 (efficiency) ，及び限界原理 (marginality principle) を満たすような値のことである．対象性によって，あるプレイヤーの解と他のプレイヤーの解が同じ価値を持つことが保証される．効率性によって，プレイヤー達は N が与える以上の値を得ることができないことを保証する．さらに限界性原理によって，プレイヤーの価値は，その提携に対する彼の貢献度によって決まることが述べられる．このように Shapley 値を特徴づける方法は，Levy らの独自の方法ではないかと推測される．ちなみにゲーム理論の教科書 [鈴木光男 81] によると，Shapley 値は Shapley によって 1953 年に”A value for n-person games ”という論文によって定義されたとなっている．この教科書によると，Shapley 値が満たさなければならない公準として，全体合理性，ナルプレイヤーのゼロ評価，対称性，及び加法性をあげている．

表 2: 性能評価の比較

DAI System	Senario 1			Scenario 2		
	Moves	Efficiency	Outcome	Moves	Efficiency	Outcome
Autonomous-Agent	10	2.95	Escape	10	2.74	Stalement
Limited-Communication	10	3.19	Escape	10	2.65	Stalement
Negotiating-Agent	11	2.32	Stalement	11	3.16	Capture
Controlling-Agent	17	1.59	Capture	12	2.62	Capture
Game-Theory	None	-	Escape	8	-	Capture
DAI System	Senario 3			Scenario 4		
	Moves	Efficiency	Outcome	Moves	Efficiency	Outcome
Autonomous-Agent	7	2.65	Stalement	10	2.23	Stalement
Limited-Communication	7	2.57	Stalement	10	2.13	Stalement
Negotiating-Agent	8	2.82	Capture	11	2.63	Capture
Controlling-Agent	10	2.07	Capture	12	2.67	Capture
Game-Theory	7	-	Capture	15	-	Capture
DAI System	Senario 5			Scenario 6		
	Moves	Efficiency	Outcome	Moves	Efficiency	Outcome
Autonomous-Agent	7	2.22	Stalement	7	2.54	Capture
Limited-Communication	7	2.22	Stalement	7	2.54	Capture
Negotiating-Agent	8	2.85	Capture	8	2.54	Capture
Controlling-Agent	10	1.89	Capture	10	2.10	Capture
Game-Theory	6	-	Stalemate	6	-	Stalemate

にいるとしよう．[15] この状況で 1 は必ず SOUTH を選択するが，2 は SOUTH または WEST を選択する．この場合もし 2 が WEST を選択すれば，2 が 1 の進路を妨害してしまい，それ以上追跡を続けることができなくなる．こういうわけで特にエージェント間の距離が 3 以内である場合には，それらの動きを調整する明確な交渉作業が必要とされると述べている．

表 2 に実験の結果を示す．この結果から次のようなことを考察している．

- [Stephens and Merx 89] で用いられた全てのエージェントが捕獲に成功したシナリオ 6 において，捕獲に失敗している．これは終盤にきて，赤いエージェントが元の場所から動いてしまっているにもかかわらず，青いエージェント達がもともと決めていた捕獲地点に向かうことを止めないからである．この論文では，通信がいくつかの均衡点 [16] の中から一つを選択する目的でのみ使われると仮定している．このシナリオにおける失敗は，間違った道を進まないように (新しい情報を手に入れる手段として) 通信が使われるべきであることを示唆している．

- 支払い関数は，全ての場合に対応するようにはつくられていない．このため，シナリオ 1, 5，及び 6 で見るような失敗を引き起こしている．一方で，青いエージェントがゲーム理論を利用して動くことによって，ほとんど通信をすることなく動き回ることができるという利点もある．

3.6 研究事例: DAI システムにおける組織的知識の表現と使用

Gasser らは，[Gasser *et al.* 89] において，分散知的システムのための一般的協調機構に向けて行なっている彼らの研究状況に関する報告を行なっている．彼らの視点において，“ 協調的枠組または組織はエージェントが他のエージェントに対して持つ信念や行為に関する既・未解決問題の特

[15] ここで，上を NORTH，右を EAST，左を WEST，下を SOUTH と表現するとする．

[16] 彼らの定義において，均衡点とは協力的提携を結んでいるエージェントが取り得る協調戦略の組の中で，最も高い効用 (utility) を得ることのできるような戦略のことである．

殊な集合体”である．また，“組織的変更は異なる問題集合を新たに設定し，個々のエージェントに対して解決すべき新しい問題を与え，さらにより重要なものとして，他のエージェントの信念や行為についての異なる仮説を与えることに相当する．”これらの考えをテストするために，彼らは知的協調実験室 (ICE) と呼ばれるテストベットを開発している．この論文では，この ICE におけるエージェントの問題表現について，追跡問題を例として議論している．

彼らの考えにおいては，組織は各エージェントがその活動の中で，何が解決されていて，何が解決されていないのか，解決されていない問題を解決するのにはどうしたら良いのか，という問題解決を行なう結果として形成されるものなのである．解決されるべき問題の質によって組織形態も自ずと異なったものとなるのである．

このような背景において，Gasser らはまずどのような問題解決が追跡問題において必要とされるのかを見るために問題分析を行ない，これが 6 フェーズからなっていると述べている．これらは，

1. 問題認識 (problem recognition) — 提示された赤いエージェントを発見する．
2. 同盟者を得る (enlisting allies) — 問題を解決するという約束のもとに青いパートナーを集める．
3. 協調の枠組を形成する (forming a coordination framework) — 行為に対する期待と約束の集合及び不均衡の解消方法を例示する．
4. 中盤の問題解決 (midgame problem solving) — 問題状態を知られている解や割り当てられた“役割”に変換する．
5. 終盤の問題解決 (endgame problem solving) — 赤いエージェントをブロックするために，知られている文脈 (例えば，下で述べる Lieb configuration) 中の小さな規則の集合に従う．
6. 終了 (termination) — 最終的なブロックした状態 (またはそれに到達できないこと) を認識する．

それぞれのフェーズは，現在のレベルの決定と行為のための文脈として，直前のフェーズで解決された課題を利用する．彼らはこれらの 6 フェーズが，数多く存在するマルチエージェント問題における解決過程というものを反映していると述べている．

彼らの定義によると，異なる組織が各フェーズやレベルで存在する．例えば，最初から赤いエージェントのどの動きを自分がブロックするかということを知っている青いエージェントはいない．すなわち，どの青いエージェントが関与するかという問題ですら未解決である．しかし青いエージェントは，協調の枠組を形成するためにどのような行為が可能か，またどのような役割が存在するかといった知識は持ち合わせている ([Cammarata *et al.* 83, Davis and Smith 83] 参照)．その後，終了状態のフェーズにおいて，どの青いエージェントがどの赤いエージェントのパスをブロックするといった役割分担がきちんとなされなければならない．これらの組織的問題を設定することにより，青いエージェントのブロックを避けるために，どのような低レベルの正確な動き及びどのような役割がとられるかといった決定を行なうような文脈が用意される．青い問題解決器は特別な問題例に対し，問題フェーズを追跡するために，いくつかの組織的遷移を行なう．

この論文では，問題点を明らかにするために，特に ICE の中盤と終盤における問題解決に注目して詳細な議論を行なっている．さらに問題を簡単にするために，彼らは Lieb configuration [17] と呼ばれるような青いエージェントの配置を考案している．この状態からゲームを始めることにより，青いエージェントが赤いエージェントをブロックすることが保証される．

[17] エージェントが次のような配置にある時，Lieb configuration と呼ばれる．まず赤いエージェントの位置を円の中心 (0,0) と考える．この中心に対して，$y = x$ 及び $y = -x$ であるような線分を考える．この時，$y > x \wedge y > -x$ であるような領域を Q_1，$y < x \wedge y > -x$ であるような領域を Q_2，$y < x \wedge y < -x$ であるような領域を Q_3，そして $y > x \wedge y < -x$ であるような領域を Q_4 と呼ぶことにする．このとき，例えば 4 つの青いエージェント B_1，B_2，B_3，及び B_4 の配置はそれぞれ，Q_1 上の (1,3)，Q_2 上の (3,1)，Q_3 上の (-1, -3)，及び Q_4 上の (-3, 0) である．

```
(Follow-Lieb-Rules ?Flr
    :Agent = __
    :RedAgent = __
    :Quadrant = __
    :Blocker = __
        c: in(?Flr.Blocker, ?Flr.Quadrant)
        m: (Generate-and-Solve
            '(QFill ?Qf
                :Agents ?Flr.Agents
                :Quadrant ?Flr.Quadrant
                :RedAgent ?Flr.RedAgent
                :Filler.Report (Send :Where ?Flr.Blocker
                                     :What ?Qf.Filler)))
    :Blocked-Position = __
        c: Next-To(?Flr.Blocker, ?Flr.RedAgent) ∧ in(?Flr.BLocker, ?Flr.Quadrant)
        m: (Generate-and-Solve
            '(Apply-LiebRules ?Alr
                :Agents ?Flr.Agents
                :Quadrant ?Flr.Quadrant
                :RedAgent ?Flr.RedAgent
                :Termination (Send :Where ?Flr.Blocker
                                   :What ?Alr.Agent)))
    Termination = __
        c: (type ?Flr.Termination) = 'Termination-Msg
        m: (Generate-and-Solve
            '(Find-and-Solve
                :From (Coord-name ?Flr.Quadrant))))
```

図 1: Lieb 戦略の記述

よって，４つの青いエージェントが割り当てられた四分円から出ないようにして，その距離を狭めることが，エージェント達がこのフェーズの問題解決において意識すべきことである．この意図に基づいて組織的関係も作られてくるはずである．そこで，Gasser らは青いエージェントに，この“割り当てられた四分円から出ないようにして，その距離を狭める”という問題を解かせようと試みている．問題表現として彼らは，Lenat らが AM [Lenat82 82] や EURISKO [Lenat 83] といった学習システムで利用したものに準拠したフレーム表現を用いている．フレーム中の各スロットには，制約条件やスロットを埋めるためのメソッド (関数) が記述される．例えば，この Lieb Configuration においてエージェントが守らなければいけない規則は図 1 に示すような問題表現として与えられる．Gasser らはこのようなフレームを既・未解決の問題表現と捕らえ，埋められていないスロットを埋めたり，また制約解消を行なうことを，エージェントが行なう一般的な問題解決手法であると位置付けている．ここでは，あるスロットの値が制約条件を見たさないとき，それは未解決の問題であり，そうでないスロットは解決された問題である．このスロット値の問題は，世界の状況の変化に連れて変化する．そこで，既に解決済みの問題であっても，状況の変化によって制約条件を逸脱し，未解決の問題となることもあり得る．

3.7 まとめ

本章では，協調のための組織的枠組を考えるための基本的問題として“追跡問題”を例として取り上げ，いくつかの研究事例について解説を行なってきた．マルチエージェントシステムにおいては，組織は各エージェントの個々の活動の結果として相互作用を行なうことによって形成されるべきものである．Gasser らのいうように，エージェントは問題解決過程において必要性に応

じて組織における役割を担うようになるかもしれない．この意味ではもはや，エージェントの組織内及び組織外活動というものを明確に区別することは困難である．

Benda らの，制御系の整ったシステムの方が問題解決能力において優れているという結論は，そのように制御系の整ったシステムを作り得る環境では，むしろ当り前の結果である．Stephens らは，目的の動的な変化というものを考慮に入れた評価を行なっており，多少進歩的である．この結果，彼らの結論の引きだし方には多少疑問を感じるが，交渉能力が柔軟な問題解決に役に立つといった見解が示されている．さらに Levy らのものになると，組織構造よりも，エージェント自身の行動決定能力というものが全面に押し出されるようになる．ゲーム理論的な手法では，組織の境界が協力的提携や非協力的提携という形で表現される．これは開放型環境では注目すべき組織観である．Gasser らの見解において，組織は，問題解決のいろいろな過程において，いろいろな抽象度で形成されるものと考えられる．このため，一つの過程においてさえ複数の抽象度の異なる組織が存在し得るのである．これは最も一般的な開放型組織の捉え方であろう．

Gasser らが追跡問題を利用した目的は，決して問題に対する効率の良い手法を提案することでもなければ，組織構造が問題解決に与える影響を議論することでもなかったのではないだろうか．彼らがこの論文で述べたかったことは，組織構造がエージェントの問題解決に影響するのではなく，エージェントの問題解決が必然的に組織構造を作り出していくのだということであろう．このような意味において，Benda や Stephens や Levy らが行なってきたような評価は，全く意味のないものではないかもしれないが，マルチエージェントシステムの持つ本質的な性質 [18] を逸脱したものとなっているかもしれないということである．このために，Gasser らは知識と言う立場からこの問題を位置付けようとしたのだと考えられる．

追跡問題は問題設定が単純であるため，エージェントの能力を考えることによってさまざまな変形が可能である．このため，組織的活動に関して何を議論したいかによって，その場に応じた設定を行ないやすい．本稿では紹介できなかったが，Durfee らによる MICE [Durfee and Montgomery 89] は囲い込み問題の発展系として興味深いものである．MICE では，赤や青のエージェントが増大した場合や，赤や青が食い合ってしまう場合なども考慮されているようである．

4 囚人のジレンマ (Prisoner's dilemma) – 利害が完全には対立しない状況での協調的行動選択 –

囚人のジレンマ (Prisoner's dilemma) [Davis 77] と呼ばれる非零和ゲームに関連したいくつかの研究を紹介することにより，利害が完全には対立しない状況での複数エージェントの協調的行動選択に関して述べる．先ず第 4.1 節で問題を定義し，第 4.2 節では問題の主性質，そしてこのゲームに関する分散人工知能の観点からの主な研究目標について述べる．第 4.3 節ではこのゲームに関してゲーム理論的解析から得られた均衡戦略について簡単に紹介する．第 4.4 節では Robert Axelrod により主催された反復囚人のジレンマ・コンピュータ選手権の結果と，それに参加した様々な戦略の諸性質について述べる．第 4.5 節では，同じく Axelrod の行った仮想的な生態学的実験に基づき，さまざま戦略の混在する集団における協調関係の発現の可能性とそれの進化に関して述べる．第 4.6 節では Genesereth らの研究を中心に，協調的意思決定における合理性の役割について述べる．最後の第 4.7 節ではこれらの諸研究から得られた結果をまとめる．

[18] これは Hewitt らが [Hewitt and Jong 84] で Open Systems という名とともに述べたように，完全な知識の共有と言うものは不可能であり，全てが共通の視点を持つことは仮定できず，また全体的な制御もあり得ないといった性質である．

4.1 問題の定義

囚人のジレンマ[19]**：** 2人プレーヤによるゲーム．各プレーヤは**協調**，もしくは**裏切り**という2つの選択肢をとることができる．互いに，相手が次にとる行動を知らないままに，自分の次の行動を選ばなくてはならない．両者が協調した場合の利得は双方に R，両者とも裏切った場合は双方に P，一方が協調し他方が裏切った場合は，協調した方に S，裏切った方に T の利得を与える．ここで各利得 T, R, P, S は $T > R > P > S$ および $R > (T + S)/2$ を満足するようにとる．このゲームを1回だけ，または繰り返し行なう場合，それぞれどのような行動選択の戦略を取るべきか？

4.2 問題の主性質および研究目的

囚人のジレンマでは，各プレーヤが2つの行動を選択でき，双方の行動選択の組合せに対して上記の制約を満たすように利得が与えられる．このようなゲームでの双方の利得は 2×2 の行列で表現することができる (例えば図 2)．一般に 2×2 のゲームは，各プレーヤにおいて，利得の大きさを相対化し (例えば利得は少ない方からそれぞれ $1, 2, 3, 4$ という相対的な値とする)，各プレーヤにおける利得の重複を許さないとすると，それは 144 種類の異なるケースに分類される．囚人のジレンマはその中の1つケースである．

囚人のジレンマにおける利得に関する最初の制約 ($T > R > P > S$) は，一方的に裏切る誘惑 T が最高で，裏切られる S が最低となることを示している．また，協調しあうときの利得 R は，裏切りあったときの利得 P より大きいことも示している．

2番目の制約 ($R > (T + S)/2$) は，両者が交互に搾取しあっても，ジレンマから抜け出せないというものである．つまり五分五分で搾取されたりする場合の平均利得が，協調しあう際の利得よりも少なくなることを示している．

囚人のジレンマでは次節で述べるように，相手がどう出ようと，自分の方は協調するより裏切った方が必ず得である．しかし，両方とも裏切りを選ぶと，両方が協調しあう場合に比べ損をしてしまうところにこのゲームがジレンマと呼ばれるゆえんがある．このゲームはチェスなどのゲームとは異なり，プレーヤどうしの利害は完全には対立しない．

囚人のジレンマを分散人工知能研究の観点からみると，**複数エージェントの利害関係がこのような性質を満たす状況における，各種の行動選択に関する考察**のための基本的問題としてとらえられる．また，囚人のジレンマでは「互いに，相手が次にとる行動を知らないままに，自分の次の行動を選ばなくてはならない」としている．これは**通信が制限されたある状況下で，複数エージェントが意思決定を行なう場合の協調の役割，そして有効な戦略**について考察するための問題であるとみなすこともできる．

このゲームは非零和非協力ゲームであり，最適戦略は存在しないことが知られている．戦略の有効性は相手がとる戦略に依存する．またゲームを1回しか行なわないのか，または繰り返し行なうのかによっても，有効な戦略への考え方が異なる．これらについて以下の各節で詳しく述べる．

---

[19]囚人のジレンマは，1950 年頃 Merrill Flood と Melvin Dresher によって発案され，後に A.W. Tucker によって定式化された．このゲームが囚人のジレンマと呼ばれるゆえんは，それが以下に述べる様な状況をもとにして説明された事情による (以下，[Hofstadter 85] からの要約)．

その状況とは，仮にあなたがもう一人の共犯者と何らかの犯罪を犯して，逮捕され，拘置されているとする．裁判を前にして二人は別々の独房に入れられている (二人の間に通信はないとする)．そこであなたのところへ検事がやってきて，ある取り引きを持ち掛ける．検事が言うには，その取り引きは共犯者にも持ちかけられているという (これは二人の犯罪者に共通の状況とする)．取り引きに関して検事が言った内容はおおよそ以下のようなことである．

「状況証拠からして，あなたが無実を主張しても，検事側はあなた達の有罪を証明してを2年の刑にする自信がある．しかしあなたが罪を認めて，あなたの共犯容疑者の罪状をあばく証言をしてくれれば，検事側はあなたを無罪放免にする．その場合，共犯容疑者には5年の刑を求刑する．ただし両方とも自白した場合は4年の禁固刑はかたいだろう.」

4.3 研究事例：戦略の均衡 – ゲーム理論的解析 –

囚人のジレンマの具体例として以下の図 2 の利得行列を考える．

プレーヤ P_1 \ プレーヤ P_2	協調	裏切り
協調	$(R=3, R=3)$	$(S=0, T=5)$
裏切り	$(T=5, S=0)$	$(P=1, P=1)$

図 2: **囚人のジレンマ** (プレーヤ P_1 の利益が左側に記されている)

１回きりの囚人のジレンマゲームにおいては，各プレーヤが自己の利益を最大化しようとすると，(裏切り，裏切り) という両プレーヤの戦略の組で均衡することがゲーム理論的分析により分かっている (例えば [鈴木光男 81])．しかし利得行列からすぐ分かるように，この均衡戦略は両者が協調しあったときよりも損になっている．このような戦略への均衡は，直感的には次のような１プレーヤの思考をたどることにより理解できる．

> もし相手が協調するなら，自分は裏切るのが適当である．もしそうすると，相手は裏切りにかわるであろう．そのとき自分は協調に変えると利得が減少するから，裏切りに留まらざるをえない．自分が裏切りに留まる限り，相手も裏切りから変えようとしないであろう．したがって，このゲームは，自分も相手も裏切ることで均衡せざるをえない．

これはゲーム理論でいうところの**ケース解析による行動決定** (相手の戦略にかかわらず，自分の利益を高める選択．後の第 4.6 節で詳しく述べる) となっている．

4.4 研究事例：反復囚人のジレンマゲームにおける様々な戦略の評価

囚人のジレンマゲームが反復される場合は，それが決まった回数しか繰り返されないとき，つまり反復回数を両プレーヤが知っている場合は相変わらず協調関係を引き出すことができない．なぜなら，最終回では両者とも (１回きりのゲームと同じ理屈で) 裏切りあうという結論が成り立つので，その前の回でも，最終回に相手が裏切るのを見越しているためにどちらも協調しようとはしない．この論法は繰り返し適用され，どんなに対戦回数が多くてもそれが有限で双方に既知であれば，この論法により前の対戦をたどっていくと最初の回まで戻ってしまう．

しかし**対戦する回数が決まっていない場合には協調関係が発現しうる**．囚人のジレンマゲームを反復して行なう場合は，**今の自分の戦略の選択が今の利益を決めるだけでなく，次回以降の相手の戦略の選択にも影響する**からである．この場合，未来の対戦の可能性は現在の戦略の選択に影響を及ぼすこととなる．

このような反復囚人のジレンマゲームにおいて，どのような戦略が有効であるかをみるために，Robert Axelrod は囚人のジレンマコンピュータ選手権を開催した [Axelrod 84]．この選手権に招待された人々は，非零和ゲームに特有な戦略的可能性について十分知り尽くしたゲーム理論の専門家達であった．彼らはこのコンピュータ選手権にプログラムを参加させた．参加者の各プログラムは，自分自身のプログラムや「でたらめ戦略」(協調，裏切りをランダムに選択) を含めてリーグ戦を行なった．各試合はちょうど 200 回の対戦からなり，各対戦の利得行列は図 2 で示したものである．表 3(次ページ参照) にこのコンピュータ選手権の参加者 (プログラム) を，表 4(最後のページ参照) にリーグ戦の各対戦結果および総合順位を示す．

表 3: **囚人のジレンマ・コンピュータ選手権の参加者** ([Axelrod 84] より)

順位	氏名	専門	プログラムステップ数	プログラムの性質	平均得点
1	An. Rapoport	心理学	4	上品で手応えがあり心が広い	504.5
2	N. Tideman & P. Chieruzzi	経済学	41	上品で手応えがあり心が広い	500.4
3	R. Nydegger	心理学	23	上品で手応えあり	485.5
4	B. Grofman	政治学	8	上品で手応えあり	481.9
5	M. Shubik	経済学	16	上品で手応えあり	480.7
6	W. Stein & Am. Rapoport	数学 心理学	50	上品で手応えあり	477.8
7	J. Friedman	経済学	13	上品だが怨み骨髄	473.4
8	M. Davis	数学	6	上品で手応えあり	471.8
9	J. Graaskamp		63		400.7
10	L. Downing	心理学	33	収益最大化原理	390.6
11	S. Feld	社会学	6		327.6
12	J. Joss	数学	5	卑劣な試し屋	304.4
13	G. Tullock	経済学	18		300.5
14	Anonymous		77		282.2
15	Random		5	でたらめ	276.3

このコンピュータ選手権で優勝した Anatol Rapoport のプログラムは「しっぺ返し」と名付けられた戦略をとるものであった．しかも驚くことに，これが最も短いプログラムであった (コンピュータチェスなどの対戦では，最も簡単なプログラムは最も弱いプログラムである場合が普通であるが) .「しっぺ返し」(英名は TIT FOR TAT) は，初回は協調し，次からは相手が前回取った行為を選択するものである．これは，相手が裏切らない限り自分から裏切ることはない (つまり上品である) が，相手が一度裏切ると，その次の回に必ず報復する (手応えがある)．しかし相手が裏切った後でも改心して協調にもどれば，それ以降は自分も再び協調する (つまり心が広い)．

以下，このコンピュータ選手権の結果から得られた，反復囚人のジレンマゲームにおける戦略の諸性質，そして教訓に関してまとめる．

- **上品さ –高得点と低得点の分かれ道–:** 高得点をあげたプログラムと低得点プログラムの戦略を比べると唯一つの性質が分かれ道となった．それは「上品さ」つまり，決して自分からは裏切らない性質のことである．成績上位の８つのプログラムはどれも上品であった．

- **収益最大化原理 –キングメーカー–:** 成績上位の８つの戦略はどれも上品であったが，その８つの戦略の順位は，下位７つの戦略のうちのある２つの戦略との対戦により大きく影響を受けた．これらの２つのキングメーカーは「収益最大化原理」に基づく戦略をとる (この戦略は発案者にちなみ「ダウニング」と呼ぶ)．この戦略は相手の行動を理解することに努め，その読みにもとづいて長期的に最善の得点を稼ぐことを意図している．これは相手に報復の気配が見えないときには，徹底して裏切って食い逃げを決め込み，逆に報復の気配が見えるときは，こちらも協調するものである．ダウニングは最初相手の協調と裏切りの確率を 1/2 と仮定して対戦を始める．よって対戦の最初の２回は「ダウニング」はどうしても相手を裏切ることになる (相手を手応えのない戦略と仮定しているから)．この相手を見積もるための試し見をすることが多くの相手に「ダウニング」に対して報復するように仕向けてしまい，結果として「ダウニング」は対戦の出だしにおける得点が悪い．１位と２位の戦略は「ダウニング」と対戦するとどちらも,「ダウニング」の方で相手を裏切るのは自分自身にとって割に合わないとみなし，協調する方が得になると判断するような対応 (下で述べる心の広い

表 4: **囚人のジレンマ・コンピュータ選手権の対戦結果** ([Axelrod 84] より)

プレイヤー \ 対戦相手	1	2	3	4	5	6	7	8
1 An. Rapoport	600	595	600	600	600	595	600	600
2 Tideman & Chieruzzi	600	596	600	601	600	596	600	600
3 Nydegger	600	595	600	600	600	595	600	600
4 Grofman	600	595	600	600	600	594	600	600
5 Shubik	600	595	600	600	600	595	600	600
6 Stein & Am. Rapoport	600	596	600	602	600	596	600	600
7 Friedman	600	595	600	600	600	595	600	600
8 Davis	600	595	600	600	600	595	600	600
9 Graaskamp	597	305	462	375	348	314	302	302
10 Downing	597	591	398	289	261	215	202	239
11 Feld	285	272	426	286	297	255	235	239
12 Joss	230	214	409	237	286	254	213	252
13 Tullock	284	287	415	293	318	271	243	229
14 Anonymous	362	231	397	273	230	149	133	173
15 Random	442	142	407	313	219	141	108	137

プレイヤー \ 対戦相手	9	10	11	12	13	14	15	平均得点
1 An. Rapoport	597	597	280	225	279	359	441	504.5
2 Tideman & Chieruzzi	310	601	271	213	291	455	573	500.4
3 Nydegger	433	158	354	374	347	368	464	485.5
4 Grofman	376	309	280	236	305	426	507	481.9
5 Shubik	348	271	274	272	265	448	543	480.7
6 Stein & Am. Rapoport	319	200	252	249	280	480	592	477.8
7 Friedman	307	207	235	213	263	489	598	473.4
8 Davis	307	194	238	247	253	450	598	471.8
9 Graaskamp	588	625	268	238	274	466	548	400.7
10 Downing	555	202	436	540	243	487	604	390.6
11 Feld	274	704	246	236	272	420	467	327.6
12 Joss	244	634	236	224	273	390	469	304.4
13 Tullock	278	193	271	260	273	416	478	300.5
14 Anonymous	187	133	317	366	345	413	526	282.2
15 Random	189	102	360	416	419	300	450	276.3

戦略) をとった．他の上品な戦略は「ダウニング」と対戦すると，ことごとく悪い方へと傾いた．

- **心の広さ –泥試合を避ける–:** 上品な戦略のうちで，前述の収益最大化原理に基づく戦略と対戦した場合に，状況を悪化させるか，しないかの対応の仕方の違いは戦略の「心の広さ」であった.「心の広い」戦略とは，相手が裏切った後でも，相手が改心して協調に転じれば，それ以降は再び協調する性質のことである．上品な戦略の中で下位から 2 番目の Friedman によるプログラムは最も「心が狭い」戦略をとるものであった．それは相手が 1 回でも裏切ると，その後ずっと裏切り続ける (つまり「怨み骨髄」) というものである．

- **相手を試す卑劣さ –裏切りの応酬を呼び起こす–:** Joss のとった「試し屋」戦略は「しっぺ返し」の焼き直しであるが，これはときおり裏切って食い逃げしようと策動する卑劣な戦略である．これは「しっぺ返し」同様，相手に裏切られると即座に必ず裏切り返すが，相手が協調した後でも 9 割協調して，1 割裏切る．つまり気まぐれにこそこそと相手の協調を食い逃

げしようとする．これは結果を大変悪くした．このような戦略をとると，その気まぐれさの確率に応じてある回数以降に裏切りの応酬が始まることがある．対戦相手が自分自身であったり,「しっぺ返し」のように報復する場合は，このようながめつい戦略は割に合わなかった．

4.4.1 反復囚人のジレンマゲームにおける教訓

以上，囚人のジレンマにおける戦略の諸性質をみてきたが，この選手権の一番の教訓は，**互いに泥試合に陥る事態を極力避けることの重要性**である．一回の裏切り行為は，それ自体の直接的効果を考えればうまくいくかもしれない．しかしその性質の真の損失は，それが果てしなき報復戦を呼び起こしてしまうことにある．この大会で成功した「しっぺ返し」が成功した要因を繰り返すと，自分から裏切り始めることはなく，相手の裏切りには即座に報復し，心が広く，そして相手に対して分かりやすい行動をとったことである．**「しっぺ返し」は相手から搾取する可能性を放棄**している．

4.5 研究事例：協調関係の発現と進化 – 仮想的な生態学的実験からの考察 –

4.5.1 協調関係が発現する必要条件

囚人のジレンマにおいて協調関係が発現する必要条件は，(1) ゲームが繰り返し行なわれ (しかもその回数が双方に分からない)，(2) かつ未来係数 (次回の利得が前回に比べて値引かれる度合) が十分に大きいことが挙げられる．未来係数が大きいほど，将来の利得が総和をとるときに大きく効いてくるので，このような場合には相互協調に向かう芽が現れてきやすい．しかしながらこれは十分条件とはならない．特に未来係数が十分大きい場合は，相手のとる戦略に関係なく，最善の結果を引き出せる戦略は存在しない (これは相手が全面的に裏切ってくる場合を考えればよい)．

4.5.2 協調関係の生態学的優勢度

それでは協調関係が発現するための十分条件は何か？またそれはあるのか？これについて考察するため，Axelrod は囚人のジレンマコンピュータ選手権に参加してきた戦略を用いて，ある種の仮想的な生態学的模擬実験を行なった [Axelrod 84]．そこでは，１回の総当たり戦を終えた後に，各戦略の得点に比例して次回 (生態学的には次世代と考えてよい) の総当たり戦における各戦略の参加者の比率を変えるというものである．これを何世代も繰り返して行ない，戦略の勢力図がどのように変化するかをみることを行なった．

この実験は，メイナード＝スミスが提唱している概念「**進化的に安定な戦略**」(evolutionarily stable strategy，略称 ESS) と深く関係する [MS 74] [MS 78]．ESS とは**個体群の大部分のメンバーがそれを採用すると，別の代替戦略によってとってかわられることのない戦略**と定義される．

さて，囚人のジレンマコンピュータ選手権に参加してきた戦略 (厳密には第２回選手権に参加した 63 の戦略) を用いて 1000 世代に渡る勢力争いのシミュレーションを行なった．結果は１回の総当たり戦の結果と同様に「しっぺ返し」が最優勢となった．また１回の総当たり戦における上位 1/3 の「上品で手応えがあり心が広い」戦略は 1000 世代目にはほぼすべてを占めるようになった．１回の総当たり戦における下位 1/3 の戦略は 50 世代でほぼ消滅した．また１回の総当たり戦では，それ自身あまり得点を挙げられない戦略と対戦して功を奏するような戦略 (「弱いものいじめ」) は，世代が進むにつれて自分が成功するのに必要な弱者を滅ぼしてしまい，結果的には自滅の道を歩んだ．

Axelrod の実験からは上記の結果が報告されているが，参加する戦略を,「しっぺ返し」を含んだいくつかの種類に限定し，さらに戦略の遺伝という機構を導入してより長期的な世代に渡って各戦略の繁栄をみるという実験の結果が [Lindgren 92] に紹介されている．この実験の結果では，必ずしも「しっぺ返し」が最良の戦略ではなく，長期的には様々な異なる戦略が繁栄/衰退を繰り返すというダイナミスクがあることが報告されている．

4.5.3 協調関係が進化する段階

仮にある集団がすべて「全面裏切り」戦略で占められているとすると，これは進化的に安定となり他の戦略が入ってくる余地がない (他の戦略が入ってくるという代わりに，ある個体の子孫が突然異変により親と異なる戦略をとるようになると考えてもよい)．何故なら，たとえそこに協調戦略が入って来たとしても，それは必ず「全面裏切り」戦略より低い得点しか得られないからである．この様な集団に協調戦略が芽生えるための必要条件については先に述べたが，この**協調戦略が繁栄するためには，その集団にいる最初は少数の協調戦略者たちが，しばらくの間一丸となって内輪つき合いを行なわなければならない**．このような状況では，協調関係は次の 3 つの段階たどって進化し，例えばこの場合，最終的には「全面裏切り」にとって代わり進化的に安定な戦略となりうる．

第 1 段階: 最初は，まわりが無条件に裏切りを繰り返すただ中で，協調関係を始めなければならない．協調戦略同士が互いに助けあう機会がない場合は協調関係の芽は育ち得ない．しかしたとえ協調戦略が少数派でも，互いに内輪でつき合う機会に恵まれ，互恵主義に基づいて協調しあうことができれば，協調は進化しうる．

第 2 段階: やがて，互恵主義に基づく戦略は，他の多くの戦略に競り勝って栄える．

第 3 段階: 互恵主義にもとづく協調関係がひとたび足場を築いてしまうと，協調的でない戦略の侵入を阻止できる (ESS への到達)．

4.6 研究事例：協調における合理性の役割

この節ではこれまでの 3 つの節とは観点を変え，様々な 2×2 の利得行列による 2 人非協力ゲームをもとに，通信をしない（もしくは通信が制限された）状況での 2 人のエージェントの相互作用について述べる（囚人のジレンマはその特殊なケースとなる）．特に，協調的相互作用へ向けてのエージェントの行為の選択や意思決定に対する合理性の役割に関して考察した Genesereth らの研究 [Genesereth *et al.* 86] について述べる．

Genesereth ら以前の DAI の研究では，複数のエージェントが同一設計者の命令の下で相互に協調することを仮定しているものがほとんどであった．つまりエージェントは競合することがなく，エージェントが複数いた場合にはそれらのゴールは同一のものか，もしくは矛盾しないものであり，エージェントは自由に相互に助け合うとしていた（このようなエージェントは**慈悲深いエージェント**と呼ばれる）．このような仮定の下で従来の DAI の研究は，(1) 行為の同期，(2) 効率の良い通信，(3) 不注意による相互干渉などの問題に焦点を当てていた．

Genesereth らは，この慈悲深いエージェントの仮定を捨て，自律的で，かつ独立した動機を持つエージェント達が，自己の目標を達成するためにどのようにして相互作用すべきかについて考察した．つまりエージェントは競合したり (ゼロサム)，無競合であったり (共通の目標)，または利害が完全には対立しないような状況に直面する．彼らは，このような状況で「なぜ合理的なエージェ

ントは他と協調することを選択するのか」とか「相互に好ましい結果をもたらすように，どのようにして行為を整合させるのか」という質問に答えようとした．さらに Genesereth らはエージェントどうしが (通信機器の故障などにより) 通信をしない状況を仮定している．ちなみに Rosenschein らの研究 [Rosenschein and Genesereth 85] ではこの仮定は外されている．

4.6.1 研究の枠組

Genesereth らは 2 人のエージェントがいて各エージェントが 2 つの行為を選択できるゲームを対象にした (以下簡単のためにこれを 2×2 ゲームと呼ぶ)．そこでは，各プレーヤにおいて利得の大きさを相対化し (例えば利得は少ない方からそれぞれ $1, 2, 3, 4$ という相対的な値とする)，さらに各プレーヤにおける利得の重複を許さないとしている．先ず準備としてエージェントの利得関数と判断手続き定義する．

エージェント i の状況 (これはある相互作用の状況を意味する) s における**利得関数** p_i^s とは，各結合動作 (つまり両者の選択) を i の効用へ写像する関数であり，

$$p_i^s : M \times N \longrightarrow R$$

となる．ここで M と N は 2 人のエージェントのそれぞれ可能な動作 (ゲームにおける手) であり，R は利得行列中のエージェント i の利得値である．利得行列は例えば図 3 のように示さる．ここで行列の各要素（四角）の中 $\setminus$ の左側の値はエージェント J の，そして右側は K の利得を意味する．

$J \setminus K$	c	d
a	$4 \setminus 1$	$2 \setminus 4$
b	$1 \setminus 3$	$3 \setminus 2$

図 3: **利得行列**

また，エージェント i の**判断手続き** W_i とは可能な状況の集合 S から，i のとりうる手の集合 M への写像で，

$$W_i : S \longrightarrow M$$

となる．

4.6.2 基本合理性

単項述語 $R_i^s(m)$ を定義する．$R_i^s(m)$ は，エージェント i が状況 s で m を手として選択することが合理的であれば真をかえす述語である．エージェント J が基本合理的であるとは，以下に示すように，エージェント J が単項述語 R を満たさない動作 m を採らないことと定義する．

$$\neg R_J^s(m) \Rightarrow W_J(s) \neq m$$

また，状況 s において，エージェント J の動作 m' が m を支配することを $D_J^s(m', m)$ と表現する．支配するとは，m' を行うことに対する利得が，m を行った場合より大きいことを示す．いま，$A_K^s(m)$ を，エージェント J が m を行ったときにエージェント K がとる動作であるとする (A は応答関数と呼ばれる)．即ち，

$$W_K(s) = A_K^s(W_J(s))$$

とする．これを用いると，**支配**することの形式的定義が以下のように与えられる．

$$D_J^s(m', m) \Longleftrightarrow p_J^s(m', A_K^s(m')) > p_J^s(m, A_K^s(m))$$

動作 m が基本合理的でないとは，以下のようにその動作を支配する他の動作 m' が存在する場合である．

$$(\exists m'\ D_J^s(m', m)) \Longrightarrow \neg R_i^s(m)$$

J が K の判断手続きを知らない場合でも，この制約は判断規則の最適さを保証し，それは**支配解析** (dominance analysis) の名で知られている．この制約にしたえば，他のエージェントのすべての動作に対して，ある動作 (例えば m とする) より高い利得を生む他の動作が存在すれば，その動作 m は禁止される．つまり，

$$(\exists m' \forall n \forall n'\ p_J^s(m, n) < p_J^s(m', n')) \Longrightarrow W_J(S) \neq m$$

基本合理性はこの支配解析を導く．

図 4 に支配解析の例を示す．この例ではエージェント J が K の動作のいかんによらず，動作 a を実行することが容易に解析できる．

$J \setminus K$	c	d
a	$4 \setminus 4$	$3 \setminus 3$
b	$2 \setminus 2$	$1 \setminus 1$

図 4: **行支配問題**

図 5 の例では単純な支配解析を適用するだけでは J の行動は決定できない．しかし，この場合は双方の基本合理性を仮定することによって動作を決定することができる．即ち，K の基本合理性を仮定することによって，動作 d が K にとっては非合理的となり排除される．従って，J は K が c を行うという前提にたって考えればよい．J にとっての残りの 2 つの可能性では，ac が bc を支配するので，b は J にとって非合理的となる．その結果，J は a を選択する．このような他のエージェントの基本合理性を仮定し判断規則の最適さを保証することは**反復支配解析**の名で知られる．基本合理性は反復支配解析も導くことができる．

$J \setminus K$	c	d
a	$4 \setminus 3$	$2 \setminus 1$
b	$1 \setminus 4$	$3 \setminus 2$

図 5: **列支配問題**

4.6.3 動作依存性

双方の基本合理性を仮定するだけでは，例えば図 6 のように行動を決定できない場合がある．このような場合には，エージェント間の動作依存性を仮定しなければ判断を行うことができない．最も単純な仮定は，エージェント間の動作は独立であるとすることである．即ち任意の m, m', n, n' に対して

$$A_K^s(m) = A_K^s(m')$$
$$A_J^s(n) = A_J^s(n')$$

$J \setminus K$	c	d
a	$4 \setminus 2$	$2 \setminus 4$
b	$3 \setminus 3$	$1 \setminus 1$

図 6: **ケース解析問題**

となる場合である．

このように仮定すると，**ケース解析**の名で知られる判断規則を適用することができる．ケース規則は「K の行動のいずれを仮定しても，J のある行動が他の行動より優っている場合には，劣っている行動を排除する」である．この**ケース解析と支配解析の違いは，ケース解析では J は K の各動作に対する 2 つの可能な動作を cross term を考慮せずに比較できること**である．

図 6 の例の場合，J は動作 a を選択すべきである．なぜなら，もし K が動作 c，d のどちらを選んだ場合でも J は a を選んだ方が利得が高いからである．この例には支配解析は適用しない．なぜならば，ad の J に対する利得は，bc の場合のそれよりも小さいからである．

基本合理性と動作独立性はケース解析を導く．さらに動作独立性と相互の基本合理性は反復ケース解析を導く．図 7 がその例にあたる．この例では，J は支配解析，反復支配解析，そしてケース解析のどれを適用しても結論は得られない．まず K についてケース解析を行うと，c が候補から外される． その仮定をもとに J は a を排除し，b を選択することができる．これらのケース解析は，エージェント間の行動の独立性と相互の基本的理性を決定することによって導くことができる．

$J \setminus K$	c	d
a	$3 \setminus 3$	$2 \setminus 4$
b	$1 \setminus 1$	$4 \setminus 2$

図 7: **反復ケース解析問題**

エージェント間の依存性としては，他に**共通振舞い** (common behavior) がある．これは，立場を代えれば J も K も同じ行動をとるという仮定である．即ち，状況 s を置換した状況を s' とすると，エージェント J と K の動作が次を満たすときに限り，この 2 人のエージェントは共通振舞いを持つという．

$$\forall s \; W_K(s) = W_J(s')$$

4.6.4 一般合理性

一般合理性は基本合理性を強化したものである．主な違いは一般合理性は個々の動作ではなく判断手続きに適用されることである．次にこれを定義する．$\mathcal{R}_i$ は手続きを引数とする単項述語で，もし引数の手続きがエージェント i にとって合理的であれば真となる．次に定義するように，一般合理的エージェントは，ある手続きがもし合理的であればそれを利用できる．

$$\neg\mathcal{R}_J(P) \Longrightarrow \exists s(W_J(s) \neq P(s))$$

判断手続きに対する合理性の定義は，個々の動作に対するそれに類似している．つまり，ある手続きはそれを支配する他の手続きがあれば非合理的であるとされる．つまり，

$$(\exists P' \; \mathcal{D}_J(P', P)) \Longrightarrow \neg\mathcal{R}_J(P)$$

となる．

ある手続きは，各ゲームで良い利得を生み，少なくとも 1 つのゲームでより良い利得を生むなら他の手続きを支配するという．$\mathcal{A}^s_K(P)$ を，もしエージェント J が手続き P を使った場合にエージェント K が状況 s で採る動作とすると，支配の形式的な定義は以下で与えられる．

$$\mathcal{D}_J(P', P) \Longrightarrow$$

$$\forall s\ p^s_J(P'(s), \mathcal{A}^s_K(P')) \geq p^s_J(P(s), \mathcal{A}^s_K(P))$$

$$\wedge\ \exists s\ p^s_J(P'(s), \mathcal{A}^s_K(P')) > p^s_J(P(s), \mathcal{A}^s_K(P))$$

一般合理性の利点は，それが共通振舞いと一緒に考慮されると，ある結合動作 (仮に xy とする) が他の結合動作 (仮に zw とする) に支配されていることがすべてのエージェントにおいて成り立つ場合，結合動作 xy をとり除くことを可能にすることである．これは**支配されたケース除去**と呼ばれる．もし結合動作が双方のエージェントにとって得にならない場合，少なくとも一方は異なる動作を行なう．支配されたケース除去により，囚人のジレンマにおける協調動作が取り扱えるようになる (図 8)．囚人のジレンマの場合においては状況は対称的（状況 s と置換状況 s' は等しい）なので，共通振舞いは 2 人のエージェントに同一動作を要求し，一般合理性は結合動作 ac が結合動作 bd 支配するので bd を除去する．

$J \setminus K$	c	d
a	$3 \setminus 3$	$1 \setminus 4$
b	$4 \setminus 1$	$2 \setminus 2$

図 8: **囚人のジレンマ問題**

残念なことに一般合理性と共通振舞いは矛盾する場合があり，図 9 に示す Battle of the Sexes と呼ばれるゲームはその例である．この場合は支配されたケース除去が共通振舞いを許さない．

$J \setminus K$	c	d
a	$1 \setminus 1$	$3 \setminus 4$
b	$4 \setminus 3$	$2 \setminus 2$

図 9: **Battle of the Sexes**

4.6.5 適用範囲

各プレーヤにおいて，利得の大きさが相対化され (例えば利得は少ない方からそれぞれ 1,2,3,4 という相対的な値とする)，利得の重複を許さない 2×2 のゲームは 144 種類考えられるが，これまでに述べた解析方法はそれらのうち 117 種類をカバーすることが報告されている．

4.7 まとめ

囚人のジレンマは一見単純なゲームであるが，最適な戦略というものは，かなり強い仮定をプレーヤにおかない限り存在しない．有効な戦略にしても，ゲームが行なわれる状況 (1 回か繰り返しおこなうか) や相手の戦略などいろいろな要因を考慮に入れなければならない．それらの要因に対して様々な関連研究が，ゲーム理論における均衡戦略にはじまり，協調における合理性の役割，

繰り返し囚人のジレンマゲームによる仮想生態学的な優位性の実験，そして協調の発現と進化などの非常に広範な立場から考察を加えている．

マルチエージェントシステムにおいては，囚人のジレンマが当てはまる状況はいろいろと考えられる．基本的には，２人のエージェントが相互作用を持つ状況で，そこで採られる行動の各効用が最初に与えた問題の定義にある制約を満たすものであれば，それは囚人のジレンマと同じ状況になる．このような明確な効用が得られない状況も多くあるが，それでも例えば以下のような条件が満たされれば，それは囚人のジレンマと同じ状況になることが知られている．

1. プレーヤどうしの利得は，互いに価値を比較できないものでもよい
2. プレーヤの利得が対等である必要はない．各プレーヤの各利得が問題の定義にある 2 つの制約を満たせばよい
3. プレーヤの利得は相対的なスケールで測ることができればよい
4. プレーヤが合理的に振舞うと仮定する必要はない
5. プレーヤのとる選択がはっきりと意識した行動でなくてもよい

このように条件をゆるめれば，囚人のジレンマに対応する状況はかなり多くなるであろう．エージェントが囚人のジレンマの状況に直面した場合 (特にその状況が繰り返される場合) は，ここで述べられたような様々な知見・教訓を考慮に入れて行動選択を行なえば，最悪の状況を避けて自己の効用，そして全体での効用というものを，より高くすることが可能かもしれない．

囚人のジレンマゲームの前提条件がそうであるように，Genesereth らの研究ではエージェントどうしは手を決定する以前に通信を行なわないことを前提としていた．しかし，Rosenschein らの研究 [Rosenschein and Genesereth 85] ではこの仮定は外されている．この場合，利害が完全には対立しないような状況で，どのような内容の情報をどの程度通信しあえば協調動作に達することができるかが興味ある研究対象となる．このようなエージェント間における情報のやりとりとそれに基づく意思決定は，一般に**取り引き**とか**交渉**などと呼ばれている．この交渉に関するスキーマの研究は，Rosenschein らの研究以降もさまざまな問題領域で盛んに行なわれており，協調に関するこのようなアプローチの研究も注目される．

4.8 謝辞

このサーベイは，1991 年 10 月の「第 1 回マルチエージェントと協調計算研究会」における会場からの要望に応える形で始められたものである．協同作業をサポート頂いた，Sony CSL の所真理雄所長および各研究員，NTT コミュニケーション科学研究所の西川清史所長，中野良平主幹研究員，並びに討論頂いた赤埴淳一氏に感謝します．

参考文献

[Agre and Chapman 87] Philip E. Agre and David Chapman. Pengi: An Implementation of a Theory of Activity. In *Proceedings of The Sixth National Conference on Artificial Intelligence (AAAI-87)*, 1987.

[Axelrod 84] Robert Axelrod. *The Evolution of Cooperation.* Basic Books Inc., 1984.

[Benda *et al.* 85] M. Benda, V. Jagannathan, and R. Dodhiawalla. On Optimal Cooperation of Knowledge Sources. Technical Report Technical Report BCS-G2010-28, Boeing AI Center,

1985. Cited in Les Gasser and Nicolas Rouquette, Representing and Using Organizational Knowledge in Distributed AI Systems. Proceedings of the 1988 Workshop on Distributed Artificial Intelligence, May 1988.

[Bond and Gasser 88] Alan Bond and Less Gasser. An Analysis of Problems and Research in DAI. In Alan Bond and Less Gasser, editors, *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., 1988.

[Bratman *et al.* 88] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and Resource Bounded Practical Reasoning. *Computational Intelligence*, Vol. 4, No. 4, pp.349–355, 1988.

[Cammarata *et al.* 83] S. Cammarata, D. McArthur, and R. Steeb. Strategies of Cooperation in Distributed Problem Solving. In *Proceedings of The Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pp. 767–770, 1983.

[Cohen and Levesque 90] Philip R. Cohen and Hector J. Levesque. Intention is Choice with Commitment. *Artificial Intelligence*, Vol. 42, No. 3, 1990.

[Davis 77] L. Davis. Prisoners, Paradox and Rationality. *American Philosophical Quarterly*, Vol. 14,, 1977.

[Davis and Smith 83] Randall Davis and Reid G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, Vol. 20, pp.63–109, 1983.

[Durfee and Lesser 88] Edmund H. Durfee and Victor R. Lesser. Predictability versus Responsiveness: Coordinating Problem Solvers in Dynamic Domains. In *Proceedings of The Seventh National Conference on Artificial Intelligence (AAAI-88)*, 1988.

[Durfee and Montgomery 89] Edmund H. Durfee and Thomas A. Montgomery. MICE: A Flexible Tested for Intelligent Coordination Experiments. In *Proceedings of the Ninth Workshop on Distributed Artificial Intelligence*, pp. 25–40, 1989.

[Gasser *et al.* 89] Less Gasser, Nicholas Rouquette, Randall W. Hill, and John Lieb. Representing and Using Organizational Knowledge in Distributed AI Systems. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence, Volume II*, pp. 55–78. Morgan Kaufmann Publishers, Inc., 1989.

[Genesereth *et al.* 86] Michael R. Genesereth, Matthew L. Ginsberg, and Jeffrey S. Rosenschein. Cooperation Without Communication. In *Proceedings of The Fifth National Conference on Artificial Intelligence (AAAI-86)*. AAAI, 1986.

[Georgeff and Lansky 87] Michael P. Georgeff and Amy L. Lansky. Reactive Reasoning and Planning. In *Proceedings of The Sixth National Conference on Artificial Intelligence (AAAI-87)*, pp. 677–682, 1987.

[Hewitt and Jong 84] C. Hewitt and P. de Jong. Open Systems. In M. L. Brodie, J. Mylopoulos, and J. W. Schmidt, editors, *On Conceptual Modeling*, pp. 147–164. Springer-Verlag, 1984.

[Hofstadter 85] Douglas R. Hofstadter. *METAMAGICAL THEMAS*. Basic Books Inc., 1985. (邦訳：竹内郁雄他訳,「メタマジック・ゲーム」，白揚社，1990 年).

[Kinny and Georgeff 91] David N. Kinny and Michael P. Georgeff. Commitment and Effectiveness of Situated Agents. In *Proceedings of The Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 82–88, 1991.

[Lenat 83] D. Lenat. EURISKO: A Program That Learns New Heuristics and Domain Concepts. *Artificial Intelligence*, Vol. 21, No. 2, pp.61–98, 1983.

[Lenat82 82] D. Lenat82. AM: Discovery in Mathematics as Heuristic Search. In *Knowledge Based Systems in Artificial Intelligence*. McGraw Hill, 1982.

[Levy and Rosenschein 91] Ran Levy and Jeffrey S. Rosenschein. A Game Theoretic Approach to Distributed Artificial Intelligence. In *MAAMAW '91 Pre-Proceedings of the 3rd European Workshop on Modeling Autonomous Agents in a Multi-Agent World (available as technical document D-91-10 of German Research Center on AI)*, 1991.

[Lindgren 92] Kristian Lindgren. Evolutionary Phenomena in Simple Dynamics. In Christopher G. Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmussen, editors, *Artificial Life II*, pp. 295–312. Addison Wesley, 1992.

[MS 74] John Maynard-Smith. The Theory of Games and the Evolution of Animal Conflict. *Journal of Theoretical Biology*, Vol. 47, pp.209–221, 1974.

[MS 78] John Maynard-Smith. The Evolution of Behavior. *Scientific American*, Vol. 239, pp.176–192, 1978.

[Pollack and Ringuette 90] Martha Pollack and Marc Ringuette. Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In *Proceedings of The Eighth National Conference on Artificial Intelligence (AAAI-90)*, pp. 183–189, 1990.

[Rosenschein and Genesereth 85] Jeffrey S. Rosenschein and Michael R. Genesereth. Deals Among Rational Agents. In *Proceedings of Nineth International Joint Conference on Artificial Intelligence*, pp. 91–99. IJCAI, 1985.

[Stephens and Merx 89] Larry M. Stephens and Matthias Merx. Agent Organization as an Effector of DAI System Performance. In *Proceedings of the Ninth Workshop on Distributed Artificial Intelligence*, pp. 263–292, 1989.

[Wilkins 88] David E. Wilkins. *Practical Planning*. Morgan Kaufmann, 1988.

[鈴木光男 81] 鈴木光男. ゲーム理論入門. 共立出版, 1981.