

複数問題の並行解決における協調手法と効率化

畑直行 藤田智之 小川均

立命館大学理工学部情報工学科

Abstract

マルチエージェントシステムにおいて、エージェント間でネゴシエーションを行い動作のステップ数に関する効率化を図ることは個々のエージェントにかかる負荷を軽減しかつ、ジョブの実行時間を短縮可能とする意味で大変重要である。しかし現実問題として考えた場合、ネゴシエーションを行わない場合に比べて必ずしも短時間で目標達成を行えるとは限らない。これはネゴシエーションにかかる時間を考慮に入れた効率の評価を行うとすると、削除された時間をネゴシエーションにかかる時間が上回る場合実行時間に関する効率化が図れたとはいえないからである。そこで本研究では、自律型エージェントがそれぞれに独立した問題を他のエージェントと同一のドメインで解決していくようなマルチエージェント環境におけるネゴシエーションについての時間的な効率化についての手法を示し、その評価を行う際にネゴシエーション手続きが有効となりうる時間的制約に関する考察と実験による検証を行った。

Cooperative and Efficient Problem Solving for Distributed Autonomous Agent

Naoyuki Hata , Tomoyuki Fujita , Hitoshi Ogawa

Department of Computer Science and Systems Engineering ,
Faculty of Science and Engineering , Ritsumeikan University

Abstract

In the multi agent environment, it is sometimes useful for each agent to reduce the total amount of the load that must be taken by their jobs to negotiate with the other one. But doing negotiation is not best way to make each agent's act efficiently in any time, since the time spent for negotiation is not usually considered in total evaluation.

So in this paper, we consider the relation of time for negotiation and efficiency of agent's acts in the case that the agents have independent problem each other, and behave as autonomous agent and work on same world. Then we argue about constrained condition such that of negotiation has to be made good use for executing total jobs of each agents in appropriate time.

1 はじめに

近年、ワークステーションや LAN などの発達により分散環境というものが注目を浴びている。このような状況の中、人工知能の分野においても分散人工知能の研究が重要な意味を持つてくると考えられる。分散人工知能とは、人間社会のような複数の自律エージェント（マルチエージェント）間における動作の相互作用を研究するものである。

マルチエージェントによる作業は、シングルエージェントのそれに比べ、処理すべき動作を分散することによって個々のエージェントの負荷を軽減することができ、処理時間の短縮化ははかれるなどの長所を持っている。しかし反面、エージェント間に競合が起こるといった問題もまた抱えている。そしてこれを解決する方法として、全てのエージェントを管理する統括エージェントを用いる方法や、競合の関係にあるエージェントの間に入り仲裁をするエージェントを用いる方法、また各エージェントが対等の立場で話し合う方法などが考えられる。そしてこれらは、エージェント間においてネゴシエーションを行うことにより競合解消が行われる。しかし、統括エージェントを用いる方法は一人のエージェントへの負荷が大きくなるなどの短所がある。そこで、本研究では各エージェントが対等の立場で話し合う方法を用いている。

これまでの研究では、ネゴシエーションを行うことによる動作ステップ数の効率化に主眼をおいたものがほとんどであり、目標が達成されるまでにかかる時間に関してはほとんど考えられていなかった。例えば、ネゴシエーションにかかるコストがあまりに大き過ぎるならば、ネゴシエーションは無意味なものになってしまうであろう。そこで本研究では、ネゴシエーションと目標達成にかかる時間に関する効率化との関係について議論する。

本研究で対象としているドメインは、同一の

世界内に存在する各自律エージェントが全て対等の立場で、かつそれぞれが他のエージェントに対して独立な目標を持つ状況下での作業である。ここで、他のエージェントのプランや目標を各エージェントが知っている場合、次に行う動作の決定や競合の回避は容易であるが、本研究では、他のエージェントのプランや目標を知らない場合においても競合を解消しながらプランの実行を行っている。

以下、2章では分散人工知能分野における問題の分類と本研究で対象とする問題について述べ、3章ではネゴシエーションと効率化についての議論を行い、4章で実験結果に基づく考察を行う。

2 マルチエージェント

2.1 分散人工知能の研究分野の分類

分散人工知能には、次に示す二つの研究分野が存在する。

- エージェント全体として一つの目標があり、それに対してそれぞれが協力してその目標を達成する。
- 全てのエージェントが個々に独立した目標を持っており、実行時に起こる他のエージェントとの処理の競合を解消しながら各自の目標を達成する。

協調問題解決を行う手法としては、大きく分けて

- 統括エージェントを用い多数のエージェントを管理し問題解決を行わせる方法。
- 契約ネットのように複数の問題を他のエージェントに依頼し分散する方法。
- 各エージェントが独自に問題解決を行おうとし、ネゴシエーションによってこれらを協調しながら、変更していく方法。

等が一般に研究されてきている。

しかし現実社会においては、このように複数の人間がある一つの問題を解決するような場合だけでなく、各自が独立した目標を持っておりそれが同一世界内にいるがゆえに目標達成過程において互いに誰かの目標達成を妨害してしまうような動作を計画する場合が頻繁に起こる。これらの問題を解決するためにエージェント同士は通信を行い、ネゴシエーションをすることによりこれらを解決していく。ネゴシエーションについては、3章で議論する。

2.2 プランニングシステム

本節では、各エージェントが分散環境においてそれぞれの目標を達成するために必要なプランニングについて議論していく。

一般的に知られているプランナとしては、STRIPS [1], NOAH [2], SIPE [4] などがある。

これらのプランナが単体で行うプランニングをシングルエージェントプランニング、プランニング機能を有する複数のエージェントが同一環境で行なうプランニングをマルチエージェントプランニングと呼ぶことにする。

シングルエージェントプランニングを行う場合は、上に示したプランナは適当であるが、複数のエージェントの協調によってプランの作成を行わなければならない(マルチエージェントプランニング)際には、共通して以下に示す問題点がある。

- (1) 作成されるプランは他のエージェントに対する影響及び他からの影響が考慮されたものではない。
- (2) 世界は静的であると仮定されている。
- (3) 他のエージェントとのコミュニケーション能力を個々のエージェントが持たない。

マルチエージェントプランニングにおけるこ

れらの問題点を解決する手段として次の二種類のアプローチが考えられる。

1. 統括マルチエージェントプランニング (*centralized multiagent planning*)
一人の統括エージェントが存在し、その他全てのエージェントはこのエージェントのもとで計画、実行を行う。例としては、契約ネット [3] が挙げられる。
2. 分散プランニング (*distributed planning*)
全てのエージェントは独立して存在し、かつ対等の立場にある。これらのエージェントが必要に応じて協調をしながらプランの実行をする。

しかし、統括マルチエージェントプランニングでは、世界全体を知るエージェントが必ず存在しなければならない、その際このエージェントにかかる負荷が大きくなるという欠点がある。

ここで、分散人工知能の観点からプランニングをとらえた際の分類を示すと、図1のようになる。

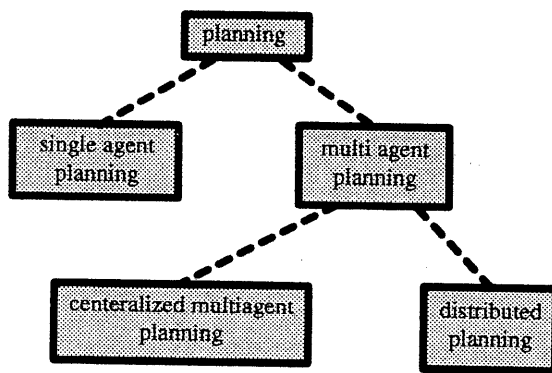


図1: プランニングの分類

なお本研究では2.1節で述べた対象に対し、分散プランニングの手法を用いることにする。

2.3 積木世界におけるプランニング

ここで、積木世界を例にとり分散プランニングを考える。例えば、図2に示すような状況を考える。この場合エージェントは二人いて、それぞれ異なる目標を持っている。それぞれのエージェントが他のエージェントの動作を考慮しなければ、それぞれの立てるプランは図3のようになる。図3における $stack(X,Y)$ は、ブロック移動のオペレータであり、ブロック X を場所 Y の上に置くという動作を示している。また、agent_2 の $stack(e,p4)$ と $stack(b,p3)$ が並列に並べられているのは、どちらのオペレータを先に実行しても目標達成を行えることを示している。

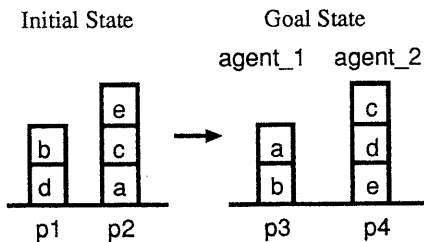


図 2: マルチエージェントでの積木世界の例

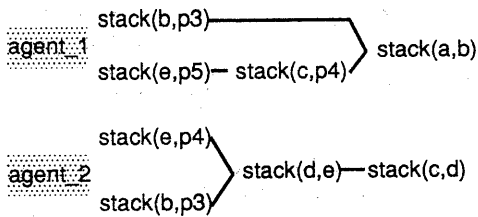


図 3: 各エージェントのプラン

しかし、この計画通りに二人が同時に動作を行う場合には、ブロックを置こうとする場所について競合が起こる。そこで、例えばお互いに

ネゴシエーションを行いながらプランの変更をし、そして実行を行うことができれば図4に示すように動作を行うことで、お互いの目標達成が可能な計画へと変更可能となる。そしてこの時 agent_1 は、動かした積木の数が $\frac{3}{4}$ となったこととなり、同様に agent_2 も $\frac{3}{4}$ ですんだといえる。

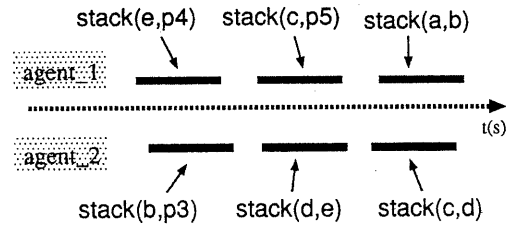


図 4: 動作結果

また別の例としては、あるエージェントがネゴシエーションを行いながらプランを行った結果、他のエージェントのみが効率化を図れて自分は効率化を図れない場合が出てくる。これは、このエージェントを親切なエージェントと設定した場合などが考えられる。しかしこの場合にも、全エージェントを平均的に見ると、目標達成の動作数に関する効率化が図れたと考えることができる。ただし、本研究においては各エージェントの効率についてのみ考え、全体的な効率についての議論は行わない。

しかし、各エージェントの目標達成の効率化を考えた時、動作のステップ数のみでなく、目標達成に要する時間についても考慮しなければならないと考えられる。つまり、目標達成を速く達成することも効率化を考える上で必要と思われるからである。そこでネゴシエーションに要する時間について考える必要がある。3章では、これについて議論を行う。

3 ネゴシエーションと効率化の関係

各エージェント間における通信が世界の状態変化の認識を可能とするだけでなく、自身の冗長な動作抑止や、エージェント間のリソース競合解消を行うことを可能とすることは前節で示した通りである。そこで各エージェント間のネゴシエーションと、動作の効率との関係を考えていく。

3.1 ネゴシエーションの手法

一般的にネゴシエーションは、対象とすべきエージェントに自分の考えを送り、その返事を基に自身が動作の変更を行うか、再度メッセージを送るという動作を繰り返すことにより行われる。図5にネゴシエーションの方法を示す。

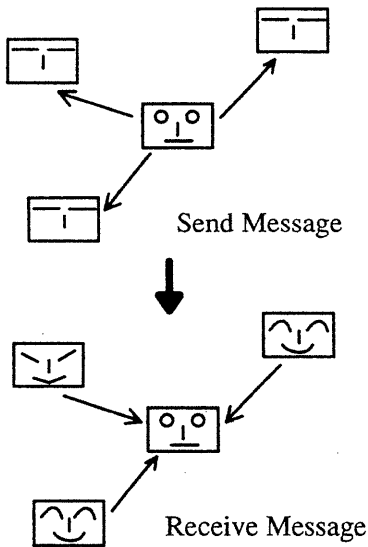


図5: ネゴシエーションの方法

相手からのメッセージをどのような判断基準で処理するかについては、一般に以下に示すよ

うな基準が考えられる。

- 基準なし（必ず提案に従う）
- 相手エージェントとの親密度
- 相手エージェントとの能力差
- 相手エージェントの提案と自分の考えとの差異

ここでこれらのどの基準が効率化を考える上で優れたものであるかという議論は本研究の範囲外である。むしろこれらの基準は、個々のエージェントの動作の効率化を考えた場合には不利となるようなものも多く含まれる。これらの基準は、エージェントモデルの研究分野で議論されている内容である。

また、ネゴシエーションの方針については以下のようなものが考えられる。

- (a) 一度のネゴシエーションにより次の動作を決定する。
- (b) 数回通信を繰り返し、意見を出し合って最終的に決定をする。

さらに、数回通信を行う場合の方法は次の二つが考えられる。

- (b-1) ある回数議論しあっても全員が納得のいく結論を得られない場合は、何らかの規則にしたがってあるエージェントの考えを決定案とする。
- (b-2) 全てのエージェントが納得がいくまで議論を続ける。

3.2 時間的な効率化

ここまでは、ネゴシエーションを行うことによる動作のステップ数の効率化について議論を行ってきた。しかし反面、ネゴシエーションを

行うことにより時間的な意味での効率が損なわれる可能性がある。つまりネゴシエーションに時間を費やすことで、結果的には目標達成が遅くなってしまう場合があるということである。そこで、どのような問題においてネゴシエーションをすべきであるのか、またどのような方法でネゴシエーションを行えばいいのかについての議論をする。

3.2.1 実行ステップ数とネゴシエーションにかかる時間の関係

各エージェントは、ある一区切りの動作（ステップ）を何回か行うことにより目標達成するとする。例えば複数のトラックでの荷物の配送計画においては、倉庫からある荷物をある場所に配送することが一ステップとなるであろう。

ここで、他のエージェントが自身の目標達成に全く影響を与えない場合を考える。例えば、他のエージェントが全く動作を行わない場合がそうである。目標達成にかかったステップ数を s_{single} 、一ステップにかかる時間を t_a 、目標達成にかかる総時間を T_{single} とすると、(1) 式で示される。

$$T_{single} = t_a * s_{single} \quad (1)$$

ここで一ステップにかかる時間 (t_a) は、一ステップの実行にかかる時間 t'_a にその動作を計画するための思考時間 t_p を加えたものである。

$$t_a = t'_a + t_p \quad (2)$$

同様にネゴシエーションを行った場合の目標達成にかかったステップ数を s_{nego} 、目標達成にかかる総時間を T_{nego} 、一ステップの動作を決定するためのネゴシエーションにかかる時間を t_n とすると、以下の式が成り立つ。

$$T_{nego} = (t_a + t_n) * s_{nego} \quad (3)$$

この式では一ステップの動作を行うたびにネゴシエーションを行う場合であるが、もし毎回ネゴシエーションを行わない場合であれば、 t_n を一ステップにかかるネゴシエーションの時間の平均値として考えれば同様に議論できるであろう。

ネゴシエーションを行うことにより得ることのできた効率を p とする。この値は、ネゴシエーションによりどれだけステップ数を減らすことができたかという値であり、以下の式が成り立つ。

$$p * s_{single} = s_{nego} \quad (4)$$

ここで p は、 $0 \leq p < 1$ の値がふさわしい。なぜなら、 $p \geq 1$ の場合は、ネゴシエーションを行うことにより目標達成に必要なステップ数が増えてしまい、ネゴシエーション基準又は方法を変更する必要性が生じる場合だからである。(4) 式を (3) 式に代入することにより、

$$T_{nego} = (t_a + t_n) * p * s_{single} \quad (5)$$

がいえる。ここで、時間の効率化の条件より、 $T_{nego} < T_{single}$ が成立する必要がある (1), (5) 式より、

$$(t_a + t_n) * p * s_{single} < t_a * s_{single} \quad (6)$$

がいえる。この式を整理することにより、

$$t_n < \frac{1-p}{p} t_a \quad (7)$$

が導かれる。

式 (7) を図示すると図 6 のように表される。これは、時間的効率化を図るために満たしていなければならない t_a と t_n との関係である。

図 6 に示されるように、斜線で表される領域がネゴシエーションが効率化を図る上で有効な領域である。例えばステップ数の効率 p が $\frac{3}{4}$ の場合、直線の傾きは式 (7) より $\frac{1}{4}$ となるため

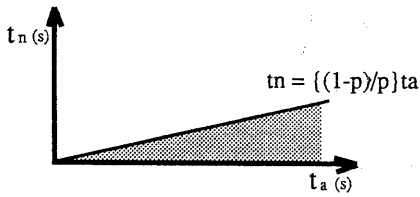


図 6: t_a と t_n との関係

に、これだけのステップ数の効率化を行うために満たされるべきネゴシエーション時間の上限が非常に小さいことがわかる。

ここで、 p は $0 \leq p < 1$ であると先に述べたが、ほとんどの場合 1 に近い値となる。これはつまり、300 回の動作をネゴシエーションにより 20 回で済ますことができる場合などほとんどないということからもわかるであろう。従って、 $t_a \gg t_n$ でなければ、時間的な効率化を図ることはできないという結果を導くことができる。

3.2.2 ネゴシエーションにかかる時間

次に、ネゴシエーションにかかる時間 t_n についてくわしく見ていくことにする。ネゴシエーションにかかる時間は次の三つに分けることができる。

時間 1 全てのエージェントにメッセージを送信してから返事が返ってくるまでの時間。

時間 2 全てのエージェントからの返事を見て自分の動作の変更を考える時間。

時間 3 該当する仕事と関係のないメッセージに対して応答する時間。

まず、(時間 1) に関しては注意しなければいけない。この時間は、基本的に自分の能力ではなく、その他のエージェントの能力に関して

くるからである。これにかかる時間としては、次の三つが考えられる。

- メッセージが伝わるまでの時間 (t_c)
- 相手のエージェントが考える時間 (t_o)
- 相手のエージェントは他の仕事をしているかもしれない、その仕事が終わるまでの時間 (t_w)

これらの値は通信の性能や相手エージェントの処理能力により、全てのエージェントで異なる。しかし、今必要な時間は全てのエージェントの中で最後に返事を返すエージェントの値である。従って、全てのエージェントにメッセージを送信してから返事が返ってくるまでの時間は、次のように表すことができる。

$$\max_{i \in E} (2t_{c_i} + t_{o_i} + t_{w_i}) \quad (8)$$

E は、メッセージの送信相手エージェントの集合である。これにかかる時間は基本的に予測不可能であり、実際にネゴシエーションを行なわせてみなければわからない。

次に(時間 2)については、一人分の返事を見て思考する時間を t_{think} とし、全てのエージェントの数を e とすると、 $t_{think} * (e-1)$ となる。これは全てのエージェントからの返事を見て自分の動作の変更を考える時間が、平均的に自分の思考スピードとエージェントの数に比例すると考えられるからである。

最後に、(時間 3) について考える。ここでは、この時間を t_r とする。これはつまり、他のエージェントの行おうとしている動作に対するネゴシエーションメッセージを送るための思考時間である。他のエージェントも各自が自身の目標達成のために考え、次に行おうとする内容をメッセージとして送信してくる。これらのメッセージの処理にかかる時間は、自身の処理能力と送られてくるメッセージの量に関係するが、

いづれだけのメッセージが送られてくるかは未知である。したがって、これらにかかる時間は一定ではなく、予測のできない値となる。また、他のエージェントからのメッセージは自分が動作を行っている間にも送られてくる場合があるが、自身の動作中に行われるこれらのメッセージに対する処理は、実行時間には影響を与えない。従って、今議論している時間 t_r に含まれてくるのは、自分は動作をしておらず、かつ自分の送信したメッセージの返事を待っている時間ではない時間のことである。

結果的に、ネゴシエーションにかかる時間 t_n は、全てのエージェントにメッセージを送信してから返事が返ってくるまでの時間と返事を見て動作の変更を考える時間、そして他のエージェントから送られてくるメッセージに対するネゴシエーションメッセージを送るためにかかる時間を合計した時間を、一ステップの動作を決めるために行うネゴシエーションの回数（メッセージを送ってその返事が返ってくるまでを一回とした時の通信回数） m で掛けたものとなる。

$$t_n = m * \left\{ \max_{i \in E} (2t_{c_i} + t_{o_i} + t_{w_i}) + t_{think} * (e - 1) + t_r \right\} \quad (9)$$

3.2.3 時間的な効率化

ここで、示してきた式より時間的な効率化を図るための手法について議論する。まず第一に結論として言えることは、式(7)を満たしていなければ、実行時間における効率化が図れないということである。先の式(7)より、 t_n は t_a に比較してかなり短くなければいけないことを示した。そこで、 p もしくは t_n を小さくすることによって式(7)を満たすことを考える。

p を小さくする、つまりステップ数における効率化を図る方法については本研究の対象外であるので、ここでは触れずにおくが、 p を小さく

くすることにより、実行時間に関する効率化も図ることができることは既知の事実である。ここでは、 t_n にかかる時間を少なくする方法について議論する。

式(9)の t_c t_w t_o t_r は、他のエージェントの動作や能力に依存する時間であるため、自身では変更できない時間である。従って、次の二つの方法が残る。

- t_{think} にかかる時間を短くする
- ネゴシエーションの回数 m を少なくする

まず t_{think} について考える。これは次に行う動作を考えるためにかかる時間であるが、大雑把に次の動作を考えるという方法に変更することなどにより時間の短縮を図ることができる。これにより、動作のステップ数が直接増えてしまう可能性も出てくるが、目標達成の実行時間に関する効率化を行う際には有効な手法であるといえる。

次の方法であるネゴシエーションの回数の減少について考える。式(9)より、 m の値を例えば4から3に減少させたとすると、 t_n の値は比例して $\frac{3}{4}$ となる。これは他の方法に比べて一番 t_n の値を減少させやすい方法であることがわかる。結果として、ネゴシエーションの回数の減少は、目標達成の実行時間に関する効率化を行う際には有効な手法であるといえる。

しかし、ここで注意すべきことは、 m と動作数に関する効率 p との間には関係があり、 m を変更することにより p の値も変わることが考えられる。しかも、これらの関係は領域に依存しており定式化を行うことは困難である。従って、ネゴシエーションの回数の減少と動作数に関する効率化の関係よりそれぞれの領域によって式(7)を満たすような値を見つけることにより、目標達成の実行時間に関する効率化を行うことができるといえる。

4 実験と考察

4.1 実験

今回の実験では、2.3節で示したような一般的な積木世界を用い、システムは、UNIX ワークステーション上で分散問題解決器 PSA [7]、非同期プロセス間通信パッケージ Mlib [7] を用いた。実験設定としては、4 エージェントが存在する世界で、各エージェントがそれぞれの独立した目標を持っている状態を考る。4 エージェントをそれぞれ1台ずつ4台のワークステーションの一プロセスとし、これらのエージェント同士の通信にはプロセス間通信を用いた。そして、このうちある一人のエージェントについての実行結果をデータとした。このエージェントは、シングルエージェントの場合には6ステップで目標達成を行えるエージェントである。また、各エージェントの動作は、実際にアームを動かすのではなくある一定の時間 wait させることによりシミュレートした。

本実験では、3章で示した式の変数を以下のように設定している。

t_a … ある一つのブロックをある場所からある場所に移動するのに要する時間（一ステップの動作にかかる時間）

t_n … ある一つのブロックを移動することを決定するのに要するネゴシエーションの時間（一ステップの動作を決定することに要するネゴシエーションの時間）

p … 他のエージェントと協調しながら動作した結果目標達成にかかったブロックの移動回数を、他のエージェントが存在しない場合の目標達成に必要なブロックの移動回数で割ったもの（ステップ数に関する効率）

T_{nego} … 結果的に目標達成に要した時間

実験結果を図7に示す。グラフには、waitの時間 (t_a) を変更しながら種々のネゴシエーシ

ンを行った結果をプロットした。縦軸が T_{nego} 、横軸が t_a 、プロットされた点の横にかかっている数字はそれぞれ (p) (t_n) である。また直線 l は、シングルエージェントの場合の t_a と T_{single} との関係であり、式 (1) より求められる。この場合、式 l は $T = 6 * t_a$ である。また、 t_n の値は、目標を達成した時にかかったネゴシエーションの総時間をステップ数で割った平均値を用いた。

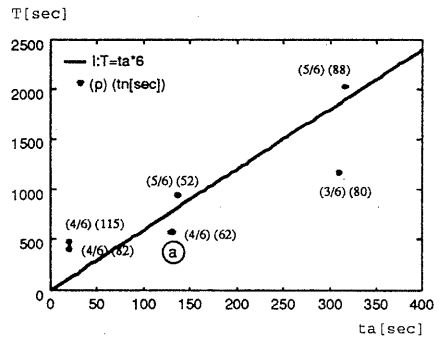


図7: 実験結果

4.2 考察

図7で、直線 l より下にあるプロットは、シングルエージェントに比べ目標達成の時間に關する効率化が行われていることになる。例えば図中のプロット a の場合には、式 (7) の左辺は、62 であり、右辺は $\frac{1-\frac{4}{6}}{\frac{4}{6}} * 130 = 65$ である。これは、式 (7) を満たしており、従って直線 l の下にプロットされている。結果として、式 (7) と図6で示した様に t_a が大きくなるにつれ効率化が行われていることがわかる。これは、 $t_a \gg t_n$ を満たしている結果となっている。

また本研究では式 (9) についての評価を行わず、 t_n に関しては単純にネゴシエーションにかかった総時間の平均値を用いたが、式 (9) のそれぞれの変数の値についての実験値による

評価をすることが必要であると考え。これにより、例えばあるエージェントのレスポンスが急に遅くなった場合にはどのようにネゴシエーションを変更すれば良いのか等の手法を考えることが可能となる。

5 おわりに

本研究では、マルチエージェントにおいての実時間を考慮に入れた効率化についての議論を行った。これらは、ネゴシエーションの基準や方法を考える上での一つの指標になると考える。そして、これらを実際にワークステーションで実験を行い、結果より導出した式の評価を行った。

今後の課題としては、本論文で示した内容の中で評価を行うことのできていないパラメータについて、さらに実験及び考察を行っていく予定である。また、今回の研究では効率化のコストを時間として議論してきたが、時間以外のあらゆるコストについても、同様に考察していく予定である。

参考文献

- [1] Richard E.Fikes , Nils J.Nilsson : STRIPS:A New Approach to the Application of Theorem Proving to Problem Solving , Artificial Intelligence, 2, pp.189-208 (1971).
- [2] Earl D.sacerdot : THE NONLINEAR NATURE OF PLANS, IJCAI , pp.206-214 (1975).
- [3] R.G.Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Trans. on Systems,

Man and Cybernetics, Vol.11, No.1, pp.61-70 (1980).

- [4] David E.Wilkins : Domain-independent Planning:Representation and Plan Generation , Artificial Intelligence, 22, pp.269-301 (1984).
- [5] P.R.Cohen / E.A Feigenbaum 編 田中 幸吉/ 淵 一博 監訳：人工知能ハンドブック第三巻，共立出版，(1984).
- [6] F.von Martial : Coordinating Plans of Autonomous Agents , Springer-Verlag , (1992).
- [7] 小川均，小嶋尚，今成文明，稲林昌二，大木宗一，井上浩一：分散型問題解決システムとその実現，信学会，AI92-51 (1992).
- [8] Victor R.Lesser : An Overview of DAI:Viewing Distributed AI as Distributed Search , 人工知能学会誌 , Vol.5 , No.4 , pp392-399 (1992).
- [9] 伊藤嘉邦，安西祐一郎：自律移動ロボットのための協調タスクプランニング，信学論 (D-II) , J75-D-II , 12 , pp.2038-2048 (1992).