

マルチステージネゴシエーションにおける探索戦略の評価

桑 原 和 宏†

マルチステージネゴシエーションにおける探索戦略(3フェーズプロトコル)の評価を行った。マルチステージネゴシエーションは分散ネットワークにおいて複数のエージェントが資源の割当を協調して行うための手法であり、そこで扱っている問題は分散制約充足問題の枠組で考えることができる。マルチステージネゴシエーションでは非同期探索フェーズ、協調探索フェーズ、過制約解消フェーズからなる3フェーズプロトコルを提案している。ここでは、分散制約充足問題の基本的なアルゴリズムの一つである非同期バックトラックとの定量的な比較を通してこの3フェーズプロトコルの性質を明らかにする。まず、マルチステージネゴシエーションで扱っている問題が一般の分散制約充足問題にどのようにマッピングされるかを明らかにし、実際の通信網のデータをもとに作成した同一の例題を用いた評価実験を行った。その結果、制約が強過ぎてすべてのゴールについて解を求めることのできない過制約の問題に対しては、どのゴールをあきらめれば残りのゴールを解くことができるかを同時に求めることもあり、3フェーズプロトコルは問題が過制約であることを認識するのに非同期バックトラックに比べ一般に時間がかかること、また、制約が強過ぎず、すべてのゴールについて解が求まるような問題に対しては3フェーズプロトコルが有効に働くことがわかった。

Evaluation of Search Strategy in the Multistage Negotiation

KAZUHIRO KUWABARA†

This paper describes evaluation results of the search strategy (which is called three-phase protocol) in a multistage negotiation. The multistage negotiation has been developed to realize resource allocation cooperatively in a distributed network. The problem it handles can be considered a distributed constraint satisfaction problem. The three-phase protocol consists of an asynchronous search phase, a coordinated search phase, and an overconstrained resolution phase. The characteristics of this protocol is clarified through comparison with an asynchronous backtracking algorithm, which is one of basic algorithms for a distributed constraint satisfaction problem. In this paper, the mapping between the multistage negotiation and the distributed constraint satisfaction is described, and evaluation results based on example problems created using a real-world communication network database are shown. The results indicate that the three-phase protocol tends to spend more time to recognize a overconstrained situation than the asynchronous backtracking algorithm due to the overhead to calculate the goals to be given up in such a situation. The results also indicate that the three-phase protocol is effective especially for problems with weak constraints.

1. はじめに

マルチステージネゴシエーションは分散ネットワークにおいて複数のエージェントが資源の割当を協調して行うための手法である^{2),3)}。マルチステージネゴシエーションはもともと通信網において通信路が故障した時、代替のパスをいかに探索したらよいかという問題を対象にしていた。そこでは通信網が複数の地域に分割され、それぞれの地域に対して割り当てられた一つのエージェントが各々の地域内の通信路の割当に責任を持つ。マルチステージネゴシエーションではどの

エージェントも完全なグローバル情報を持たず、各エージェントは自分の局所的な資源の割当を他のエージェントに通信し、お互いにその影響を交換しあうことによって全体として整合のとれた資源の割当を行う。

これに似た枠組として契約ネットワーク^{4),10),11)}が提案されている。契約ネットワークはタスクのアナウンス、入札、落札を通して複数のエージェント間でタスクの割当を実現するものである。契約ネットワークでは各エージェント独自の判断で入札、落札を行うため、タスクの割当においてグローバルな制約がある場合、全体として整合性のとれた割当を実現するのが難しいという問題点がある。これに対し、マルチステージネゴシエーションでは局所的な資源割当

† NTT コミュニケーション科学研究所
NTT Communication Science Laboratories

の影響を繰り返して交換するため、分散資源割当においてその割当の間でグローバルな制約がある場合でも全体として適当な割当を実現することができ、契約ネットワークプロトコルの一つの拡張として考えることもできる²⁾。

このようなマルチステージネゴシエーションではエージェント間で資源の割当およびその影響に関する情報の交換を行う必要があり、どのような情報を交換したらよいかマルチステージネゴシエーションにおける課題の一つであった⁷⁾。さらにエージェント間で交換した情報に基づいてエージェントがどのように振舞うべきかが問題であり、マルチステージネゴシエーションではその探索戦略として3フェーズプロトコル^{2),5),6)}を提案した。このプロトコルはマルチステージネゴシエーションで定義されているグローバルゴール、グローバルプランなどの概念に基づいており、このプロトコルを一種の探索戦略と見なした場合どのような性質を持つか、十分な定量的評価がなされていなかった。そこで、マルチステージネゴシエーションで扱っている問題をより一般的な枠組でとらえなおし、その枠組で用いられている汎用のアルゴリズムと性能を比較することにより3フェーズプロトコルの性質を明らかにしたい。

一方、マルチステージネゴシエーションで扱っている問題は制約充足問題⁹⁾が複数のエージェントに分散している分散制約充足問題と考えることもできる^{14),15)}。そこで、ここではマルチステージネゴシエーションで扱っている問題を分散制約充足問題へマッピングし、同一の問題をマルチステージネゴシエーションおよび分散制約充足問題としてとらえ、各々の手法で解いてみることにより、3フェーズプロトコルの定量的な性質を比較・評価する。

ここでは次のような評価方針を立てた。

- 分散制約充足問題ではいくつかの解法^{1),14),15)}が提案されているが、その中の基本的な解法の一つである非同期バックトラック^{14),15)}との比較を行う。このアルゴリズムを選んだのはこのアルゴリズムをベースに性能評価例が報告されており⁹⁾、分散制約充足問題に

において比較の基盤を与える基本的なアルゴリズムの一つであると考えられるからである。

- 実際の応用に近い問題で評価するため NTT の通信網のデータより作成した通信網のパスの設定問題を例題とする。

本論文の構成は以下のとおりである。まず、マルチステージネゴシエーションとその探索戦略の概略を述べ、次に分散制約充足問題とその解法である非同期バックトラックの概略を述べる。さらに、マルチステージネゴシエーションで扱われている問題が分散制約充足問題にどのようにマッピングされるかを述べる。そして、性能比較の実験の結果を基にマルチステージネゴシエーションと非同期バックトラックの両者の性能を比較し、その得失を議論する。

2. マルチステージネゴシエーション

2.1 問題の表現⁷⁾

2.1.1 探索空間

マルチステージネゴシエーションの探索空間は図1に示すように表現できる。いわば全体として解くべき目標であるグローバルなゴールがいくつか存在する。このグローバルゴール g_i はグローバルゴールを満足するためのエージェントにまたがる仮想的なプランであるグローバルプラン $gp_{i,j}$ の集合を持つ。あるグローバルゴールに対してグローバルプランのうち、一つでも満足されるとグローバルゴールが満足されることになる。グローバルプランはエージェントが局所的に解けるいくつかのサブゴールに分解される。 $sg_{i,j,k}$ がグローバルプラン $gp_{i,j}$ に対する k 番目のサブゴールを表す。サブゴールはある唯一のエージェントに属するものである。一つのサブゴールがいくつかのグ

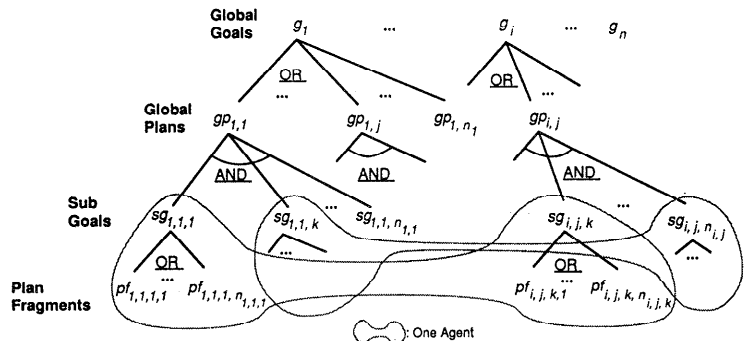


図1 マルチステージネゴシエーションにおける探索空間⁷⁾
Fig. 1 Global search space in the multistage negotiation⁷⁾.

ローバルプランに同時に属する場合もある。エージェントはそのサブゴールを満足するためのいくつかの方法（ここではプランフラグメントと呼ぶ） $pf_{i,j,k,l}$ を持つ。プランフラグメントはある唯一のサブゴールに属するものとする。

これらの関係は論理表現を用いると次のようになる。

$$g_i \Leftrightarrow \bigvee_j gp_{i,j}$$

$$gp_{i,j} \Leftrightarrow \bigwedge_k sg_{i,j,k}$$

$$sg_{i,j,k} \Leftrightarrow \bigvee_l pf_{i,j,k,l}$$

ここで g_i , $gp_{i,j}$, $sg_{i,j,k}$, $pf_{i,j,k,l}$ はそれぞれが満足される、または選択される時に真、そうでない時に偽の値をとるものとする。

2.1.2 制 約

資源の使用に関する制約とすべてのグローバルゴールを満足しなければならないといういわば暗黙の制約の二種類の制約がある。

前者の資源の使用に関する制約はさらにエージェント内に閉じた資源使用の制約（エージェント内制約）、エージェントにまたがった資源使用の制約（エージェント間制約）の二種類に分けて考えられる。

- エージェント内制約：プランフラグメントを実行するためには一定の資源を確保する必要があり、資源の量には限りがあることから同時に選択できるプランフラグメントの組には制限がでてくる。形式的にはプランフラグメントの集合 PF が同時に選択できるためには

$$\sum_{pf_u \in PF} needs(pf_u, r_v) \leq copies(r_v) \quad (\text{for all } v)$$

の条件を満たす必要がある。ここで $needs(pf_u, r_v)$ は pf_u の実行に必要な資源 r_v の数、また、 $copies(r_v)$ は資源 r_v のエージェントにおいて使用可能な数を表す。ここではプランフラグメントはある一つのサブゴールに対応しているという仮定から、この制約により、エージェント内で満足できるサブゴールの組に制限があることになり、このようなサブゴールの組がエージェント内制約となる。

- エージェント間制約：複数のエージェントに関係する資源使用の制約であり、例えばエージェント A で資源 ra が使用される時、 B では資源 rb が使用されなければいけないというようなものである。この時、いずれか一方の資源のみを使用する

ということは制約違反となる。すなわち、エージェント A と B で資源の使用に関して制約があり、エージェント間においてこのような資源の使用の制約を満足するように同時に選択されなければならないプランフラグメントの組が存在することになる。このためエージェントにおいて満足されるべきサブゴールの組が存在することになり、このようなサブゴールの組がエージェント間制約となる。なお、エージェント間制約が探索空間におけるグローバルプランに対応することになる。

また、後者のグローバルゴールを満足しなければならないという制約（以下グローバル制約と呼ぶ）は具体的には各々のグローバルゴールについて必ず一つのグローバルプランが選択され、それを構成するサブゴールすべてが満足されなければならないということである。

マルチステージネゴシエーションは以上述べた探索空間において上の制約を満たすように満足すべきサブゴールとそのために必要なプランフラグメントを選択する問題として考えることができる。

2.2 競合関係の表現⁷⁾

探索を進めていく上でグローバルゴール、グローバルプラン、サブゴール間の競合関係を各エージェントが認識することが必要である。マルチステージネゴシエーションでは次のように競合関係を表現する。

2.2.1 Nogood Goal Set

グローバルゴール間の競合関係を示したものが Nogood Goal Set である。Nogood Goal Set は論理表現を用いて形式的には次のように表現できる。

$$\bigvee \bigwedge \neg g_i$$

Nogood Goal Set が求められればグローバルゴール間の関係がわかり、例えば制約が強過ぎるような場合にどのグローバルゴールをあきらめればよいかわかる。もし、同時にすべてのグローバルゴールが満足できるのであれば Nogood Goal Set は空となる。

2.2.2 Goal Exclusion Set

ここでは一つのグローバルゴールに対してその解を求めることに責任を持つ起動エージェントというものを仮定しており、起動エージェントはその責任を持つグローバルゴールを p-goal として扱う。起動エージェントは自分の p-goal に関して他のグローバルゴールのグローバルプランとの競合関係（これを Goal Exclusion Set と呼ぶ）を求め、それを交換しあうことで Nogood Goal Set を求める。ここでグローバル

ゴール g_i とグローバルプラン $gp_{j,k}$ ($j \neq i$) 間の競合関係 (Goal Exclusion Set) は形式的には次のように表現できる。

$$\neg g_i \vee (\bigvee_{j \neq i} \bigwedge \neg gp_{j,k})$$

2.2.3 Exclusion Set

グローバルゴールの Goal Exclusion Set はそのグローバルに含まれるサブゴールの Exclusion Set から計算される。このサブゴール $sg_{i,j,k}$ の Exclusion Set とはそのサブゴールと競合関係にあるグローバルプランの集合であり、これは形式的には次のように表現できる。

$$\neg sg_{i,j,k} \vee (\bigvee_{l \neq i} \bigwedge \neg gp_{l,m})$$

2.2.4 Conflict Set

サブゴールの Exclusion Set は、同一のエージェント内でサブゴールと競合関係にあるサブゴールの集合 (これを Conflict Set と呼ぶ) から計算する。エージェント内でのサブゴール $sg_{i,j,k}$ の競合関係は形式的には次のように表現できる。

$$\neg sg_{i,j,k} \vee (\bigvee_{l \neq i} \bigwedge \neg sg_{l,m,n})$$

これはサブゴールを満足するためのプランフラグメントの資源の要求条件からエージェント内だけで計算できるもので、エージェント内制約に対応するものである。

2.3 探索戦略^{(2), (5), (6)}

マルチステージネゴシエーションにおける探索戦略として非同期探索、協調探索、過制約解消の三つのフェーズからなる3フェーズプロトコルを導入している (図 2)。ここでの基本的な考え方はまず複数のグロー

バルゴールに関してばらばらに解を求めようとして、すべての解が求まればよしとし、もし、求まらなければ前述のエージェント間の競合関係をすべて計算するというものである。さらにここでは、NoGood Goal Set が空でなく、制約が強過ぎることがわかったならば適当なグローバルゴールをあきらめて残りのグローバルゴールのみを満足させるという方針をとっている。

2.3.1 非同期探索フェーズ (Asynchronous Search Phase)

このフェーズでは起動エージェントはその p-goal を満足させる解の探索を始める。もし、すべてのグローバルゴールが満足されたならばネゴシエーションは終了する。

ここではエージェントはまず仮にサブゴールを選択し、そのサブゴールを満足するためにプランフラグメントを選び、それが必要としている資源を調べる。エージェント間制約が存在する資源を持つエージェントに対し、その資源使用の是非を確認するメッセージ (ok? メッセージと呼ぶ) をそのサブゴールのグローバルプランの情報とともに送る。メッセージを受け取ったエージェントはエージェント間制約を満たすようにサブゴールを選択できるか調べ、可能ならば ok メッセージを返し、だめならば conflict メッセージとともにその競合情報、すなわち、ok? メッセージを受け取った側のエージェントのサブゴールの Exclusion Set を送る。このようにして、このフェーズでは探索と同時に Exclusion Set の情報が次第に集められる。

エージェントが同じサブゴールを二度目に選択する時に前と同じ状況で試みても意味がないので他のエージェントの選択を強制的に変更させる必要がある。これがバックトラックに相当する。これは retry メッセージにより実現され、一度選択されたサブゴールのプランフラグメントに対して割り当てられた資源を解放して、要求されたサブゴールを満足しようとする。

2.3.2 協調探索フェーズ (Coordinated Search Phase)

このフェーズでは起動エージェントは Nogood Goal Set を計算する。そのために p-goal の Goal Exclusion Set を完全にする。ここで完全とはこれ以上値が変化しなくなるということで、この判断は他のエージェントにおける Exclusion Set が完全になったという通知によって行われる。Goal Exclusion Set が起動エージェント間で交換されることによって、Nogood Goal Set が求められ、同時に複数のグロー

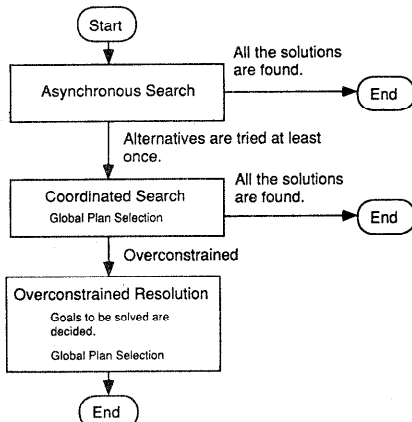


図 2 3フェーズプロトコル⁽⁵⁾
Fig. 2 Three-phase protocol⁽⁵⁾.

バルゴールが満足されないことがわかった時は制約が強過ぎるということなので過制約解消フェーズへ移行する。

2.3.3 過制約解消フェーズ (Overconstrained Resolution Phase)

このフェーズでは前のフェーズで求められた No-good Goal Set に基づいてあきらめるべきグローバルゴールを決定する。あきらめるゴールについて合意が得られた時は残りのグローバルゴールに関してどのグローバルプランを用いて解くかを Goal Exclusion Set の情報に基づいて決定する。この段階で Goal Exclusion Set, Exclusion Set にはサブゴールに関して競合情報が集められているので、バックトラック無しに解を見つけることができる。

3. 分散制約充足問題

3.1 問題の表現

一般に制約充足問題は m 個の変数 x_1, x_2, \dots, x_m とそれらが取りうる値の領域 D_1, D_2, \dots, D_m および、その上で定義された制約 $P_k(x_{k_1}, x_{k_2}, \dots, x_{k_j})$ が与えられた時にすべての制約を満たすように変数の値を決める問題と定義できる。分散制約充足問題は変数、制約が複数のエージェントに分散された問題として定式化できる。すなわち、変数があるエージェントに割り当てられ、エージェントは割り当てられた変数の値の決定を行う。

3.2 非同期バックトラックアルゴリズム

分散制約充足問題のアルゴリズムはいくつか提案されているが、ここではその中でも基本的なものの一つである非同期バックトラックのアルゴリズム¹⁵⁾を取り上げる。このアルゴリズムの基本的な考え方は次のとおりである (ここでは文献 15) にならい一つのエージェントに対し一つの変数が割り当てられると仮定する)。まず、このアルゴリズムでは次のような前提条件を仮定している。

- エージェントに対してユニークな識別子 (id) が振られており、全順序がつけられている*。より小さい id を持つエージェントを高位のエージェントとする。
- 制約に対しそこに含まれる変数のエージェントのうち一番 id が大きいエージェントがその制約を評価する。制約に含まれる変数のエージェントか

* 例えばエージェントの id を名前と見なして、その辞書式順序をとればよい。

ら制約を評価するエージェントへ向かって論理的なリンクが張られている。

エージェントは自分に向かって張られたリンクを持つ他のエージェントの局所解の組合せ (Agent_View) と制約違反を起こす変数と値の組合せ (NoGood) を保持する。これらの値は最初は空である。

各エージェントは Agent_View および NoGood と矛盾しないように自分の変数の値を決めその値 (局所解) を外に向かって張られたリンク先のエージェントに対し送る (ok? メッセージ)。このメッセージを受けとったエージェントは Agent_View を更新し、この新たな Agent_View と矛盾しないように自分の変数の値を決め、変更した時はその値 (局所解) を外に向かって張られたリンク先のエージェントに対し送る。もし、そのような変数の値が決められない時はその原因となる変数と値の組合せ (NoGood) を計算し、その NoGood 中の一番大きい id を持つエージェントに対し、その NoGood を送信し (NoGood メッセージ)、変数の値の変更を要求する。NoGood メッセージを受けたエージェントは自分の NoGood を更新し、新たに自分の変数の値を変更する。もし、それが不可能な場合はさらにその原因となる変数と値の組合せ (NoGood) を計算し、その NoGood 中の一番大きい id を持つエージェントに対し、その NoGood を送信する。NoGood は単調に増加するので無限ループを避けることができる。

一つのエージェントに複数の変数が割り当てられる場合はエージェントに対して id を割り振るのではなく、変数に対して id を割り振り、変数単位に上で述べたアルゴリズムを動かせばよい。この時エージェント内でのメッセージのやりとりは手続きの呼び出しになる。また、NoGood に関しては単調に増加するので、エージェント内に一つだけ設け、エージェント内のすべての変数から共通にアクセスされるようにすることができる。

4. マルチステージネゴシエーションの分散制約充足問題へのマッピング

マルチステージネゴシエーションではグローバルゴール、グローバルプランといった概念を扱っている。そのため全体で複数のゴールがいくつかの方法で満足されるような問題のマッピングは自然な形で行うことができる。さらに制約が強過ぎる時にどのグローバルゴールをあきらめればよいかなどアプリケーション

ンから見て意味のある情報の抽出を解法の中に含めることが容易にできる。

しかし、それらのグローバルゴールなどの概念の意味を無視し、サブゴール間の関係を制約として表現することで、マルチステージネゴシエーションで扱っている問題を分散制約充足問題としてとらえることができる。この時、マルチステージネゴシエーションの問題を分散制約充足問題の一般的な形式にどのようにマッピングしたらよいかを明らかにする必要がある。

分散制約充足問題は変数とその領域、制約、さらに変数がどのようにエージェントに分散されるかによって定義される。したがってマルチステージネゴシエーションの問題を分散制約充足の変数、領域、制約、エージェントにどのようにマッピングしたらよいかを決めればよい。

4.1 変数とその領域

変数にはマルチステージネゴシエーションにおけるサブゴールを対応させる。その場合の変数の領域はサブゴールを選択しない(これを NIL で表すとする)か、選択する時はそのサブゴールに含まれるどのプランフラグメントを選択する(選択されるプランフラグメントをその値とする)かになる。すなわち、サブゴール $sg_{i,j,k}$ に対して

$$sg_{i,j,k} \Leftrightarrow \bigvee_l pf_{i,j,k,l}$$

という関係がある時、これに対応した変数を $X_{sg_{i,j,k}}$ とするとその領域は

$$\{NIL, pf_{i,j,k,1}, \dots, pf_{i,j,k,i,j,k}\}$$

となる。

これにより変数の値の選択はマルチステージネゴシエーションにおけるエージェント内で行うことができるのでエージェントの対応は分散制約充足とマルチステージネゴシエーションとで同じにすることができる。

4.2 制 約

制約としては資源の使用に関する制約とさらにすべてのグローバルゴールが満足されなければならないというグローバル制約がある。

前者の資源使用に関する制約はエージェント内制約とエージェント間制約に分けられ、それぞれ次のように対応付けられる。

- エージェント内制約：資源の容量の制約によりエージェント内で選択できるプランフラグメントに制限ができ、そのため、エージェント内で満足できるサブゴールに制約が生じることである。こ

れは同時にはサブゴールに対応する変数の値が NIL でない値がとれない組合せがあるということになる。ここで $V_{A,t}$ をエージェント A において同時には満足できないサブゴールの集合を表すとする。t はそのような集合が複数あることを示すための添字である。これを用いるとエージェント内制約は形式的には次のように書ける。

$$\neg \bigwedge_{sg_{i,j,k} \in V_{A,t}} (X_{sg_{i,j,k}} \neq NIL)$$

- エージェント間制約：エージェント間において資源の使用に制約があるため、あるサブゴールを満足させた時に同時に満足させなければいけない(他のエージェントにおける、同じグローバルプランに属する)サブゴールがあることである。エージェント間制約があるサブゴール $sg_{i,j,k}, sg_{i',j',k'}$ 間の制約は形式的には

$$\neg((X_{sg_{i,j,k}} \neq NIL) \oplus (X_{sg_{i',j',k'}} \neq NIL))$$

と書ける。ここで \oplus は排他的論理和を表す。ここで排他的論理和を使ったのは両者のサブゴールが同時に満足されるかまたは満足されないかのどちらかであるからである。

また、すべてのグローバルゴールが満足されなければならないというグローバル制約はすべてのグローバルゴールに関してどれか一つのグローバルプランが満足されるということである。あるグローバルプランのサブゴールのうち満足されるサブゴールが存在すれば上記のエージェント間制約によりそのサブゴールが属するグローバルプランは満足されることになる。したがってグローバル制約は形式的には次のように書ける。

$$\exists j \exists k X_{sg_{i,j,k}} \neq NIL \text{ (for all } i)$$

5. 通信網のデータを用いた評価実験

ここでは、実際の応用に近い問題で評価するため NTT の通信網のデータより作成した通信網のパスの設定問題を例題としてとりあげた。評価実験の例題を述べる前にまず、簡単な通信網の例をとりあげ、それがどのようにマルチステージネゴシエーションと分散制約充足問題として考えることができるかを示す。

5.1 簡単な通信網における両者へのマッピング

図 3 に通信網の例を示す。ここでは A, B, C, D, E の五つの地域 (Region) に分かれており、中継点 (Station) として 1a, 1b, 1c, 2c, 1d, 1e がそれぞれ地域内にあり、また、ac, bc, cd, ce がそれぞれ地域間の境界点として存在する。図に示すように中継

点間に通信路 (チャネル (Channel)) が存在し、これが資源 (Resource) に相当する。それぞれ容量は1であるとする。ここでのゴールは 1a と 1d 間、1b と 1e 間にそれぞれ容量1のバスを確保することである。この例では資源 (通信路) $r3$ が両者のゴールを満足させるのに必要なため、両方のゴールを同時には満足できない*

5.1.1 マルチステージネゴシエーションへのマッピング

上で述べた問題はマルチステージネゴシエーションでは次のように考えることができる。まず、それぞれの地域がエージェントに対応する。グローバルゴールはある地点からある地点へのバスを設定することであり、グローバルプランは表1のような中継点を通るものとなる。次に各エージェントにおけるサブゴールは表2に示すようになる。また、表3は各エージェントにおける資源とその総量、ならびにサブゴールに対するプランフラグメントとプランフラグメントが使用する資源の量を示す。この例では一つのサブゴールに対して一つのプランフラグメントが存在している。

ここでのエージェント内制約とは各通信路の容量の制約に基づくものである。エージェントCにおいて

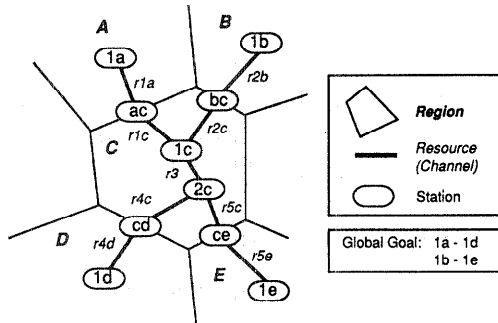


図3 例における通信ネットワーク
Fig. 3 Communication network example.

表1 例におけるグローバルプラン
Table 1 Global plans in the example.

Global Goal	Global Plan	Stations Used
$g1$	$gp1$	1a-ac-1c-2c-cd-1d
$g2$	$gp2$	1b-bc-1c-2c-ce-1e

* 実際の通信網では $r3$ を二重化するなどの冗長構成をとることが考えられるが、ここでの例題はあくまでもマルチステージネゴシエーションと分散制約充足問題へのマッピングを説明するために非常に簡単化したものである。

ランフラグメント $p1c$ と $p2c$ はともに資源 $r3$ を使用するため同時には選択できない。したがって、サブゴール $1c$ と $2c$ は同時には満足できないというエー

表2 例における各サブゴール
Table 2 Subgoals in the example.

Agent	Global Goal	Global Plan	Subgoal	Stations to be Connected
A	$g1$	$gp1$	1a	1a-ac
B	$g2$	$gp2$	1b	1b-bc
C	$g1$	$gp1$	1c	ac-cd
	$g2$	$gp2$	2c	bc-ce
D	$g1$	$gp1$	1d	cd-1d
E	$g2$	$gp2$	1e	ce-1e

表3 例における各エージェントのサブゴール、プランフラグメント、資源

Table 3 Subgoals, plan fragments, and resources in the example.

Agent A		Resource				
Subgoal	Plan Fragment	$r1a$				
Resource Count		1				
1a	$p1a$	1				
Agent B		Resource				
Subgoal	Plan Fragment	$r2b$				
Resource Count		1				
1b	$p1b$	1				
Agent C		Resource				
Subgoal	Plan Fragment	$r1c$	$r2c$	$r3$	$r4c$	$r5c$
Resource Count		1	1	1	1	1
1c	$p1c$	1		1	1	
2c	$p2c$		1	1		1
Agent D		Resource				
Subgoal	Plan Fragment	$r4d$				
Resource Count		1				
1d	$p1d$	1				
Agent E		Resource				
Subgoal	Plan Fragment	$r5e$				
Resource Count		1				
1e	$p1e$	1				

エージェント内制約が存在することになる。

また、通信路という性格から地域間の境界点に達する通信路はその境界点の両側でそれぞれ通信路が割り当てられる必要がある。通信路が資源に相当することから、境界点をはさんで資源が同時に使用されるというエージェント間の制約があることになる。エージェント間の資源使用の制約からサブゴール間の制約が生じ、この例題ではエージェント間制約として、次のサブゴールの組はどちらか一方が満足されたら残りのサブゴールも満足されなければならないことになる。

(1a, 1c), (1c, 1d), (1b, 2c), (2c, 1e)

5.1.2 分散制約充足問題へのマッピング

先の例題の分散制約充足問題へのマッピングを行う。

- 変数：マルチステージネゴシエーションにおけるサブゴールに対応して存在する。
- 変数の領域：選択すべきプランフラグメント（この例では各サブゴールに対して一つ決まる）か、何も選択しない (NIL) かのどちらかになる。
- 制約：次のような種類の制約になる。

—エージェント内制約：プランフラグメント $p1c$ と $p2c$ は同時には選択できないため、エージェント C においてサブゴール $1c$ と $2c$ は同時には満足できない。したがって、エージェント内制約は表 4 に示すようになる。

—エージェント間制約：エージェント間で隣接する通信路が接続されなければならないという制約に対応し、具体的には表 5 に示すようになる。

—グローバル制約：各グローバルゴールに関してグローバルプランは一つずつ存在する。それらのグローバルプランから選んだサブゴールの一

表 4 例におけるエージェント内制約
Table 4 Intra-Agent constraints in the example.

Agent	Constraint
C	$\neg((X_{1c} \neq \text{NIL}) \wedge (X_{2c} \neq \text{NIL}))$

表 5 例におけるエージェント間制約
Table 5 Inter-Agent constraints in the example.

Global Plan	Constraint
$gp1$	$\neg((X_{1d} \neq \text{NIL}) \oplus (X_{1c} \neq \text{NIL}))$ $\neg((X_{1c} \neq \text{NIL}) \oplus (X_{1d} \neq \text{NIL}))$
$gp2$	$\neg((X_{1b} \neq \text{NIL}) \oplus (X_{2c} \neq \text{NIL}))$ $\neg((X_{2c} \neq \text{NIL}) \oplus (X_{1b} \neq \text{NIL}))$

つに対応する変数が NIL 以外の値をとらなければならない。この時、通信路の始点のサブゴールを選ぶことによってグローバル制約は始点のエージェントの中に閉じた制約とすることができる。例えば表 6 に示すようになる。

5.2 評価用例題の作成

ここでは NTT 伝送システム研究所で開発されたネットワーク構成管理データベース (CMDB: Configuration Management Database)^{12),13)} のデータを利用して例題を作成した。このデータベースには NTT における通信網を反映した約 20 万本の通信路 (64K 回線相当で約 60 万回線分のデータ) が登録されている。

このデータベースのデータより情報を抽出し、性能評価用の問題を作成した。ここではまずマルチステージネゴシエーションの枠組に基づいた問題を作成し、次に第 4 章で述べた手法にしたがって分散制約充足問題への変換を行った。使用したデータは基本的には交換局 (中継点に相当) とその局間を結ぶ通信路のデータである。ここで作成した問題は前章で述べた例題を複雑にしたものであり、例題の作成方針は以下のとおりである。

- エージェント：日本全国 10 の地域に分割し、各地域にエージェントを割り当てた。具体的には北海道、東北、関東、信越、北陸、東海、関西、中国、四国、九州の地域に分けた*。
- グローバルゴール：通信路として 5760 回線の容量を持つ通信路を用いて各地域を代表する交換局を結ぶパスを求めることをグローバルゴールとする。各々の地域の中に代表となる交換局は単純にデータベース中の id の一番若いものを選んだ。ここで 5760 回線容量のものを選んだのはネットワークにおける主要なバックボーンのパスの設定を行う問題とするためである。

また、10 の地域に分かれることから、 ${}_{10}C_2=45$ 個のグローバルゴールが考えられる。この中からそれぞれの問題について異なる三つのグローバル

表 6 例におけるグローバル制約
Table 6 Global constraints in the example.

Global Goal	Constraint
$g1$	$X_{1a} \neq \text{NIL}$
$g2$	$X_{1b} \neq \text{NIL}$

* この地域は東京支社を関東に含めた以外は NTT の各支社に対応している。

ゴールを設定した。

- グローバルプラン：パスの候補がグローバルプランに相当する。グローバルプランとしては代表点間を結ぶパスにおいて通過する交換局の数が最小となるものすべてを選んだ。
- サブゴール，プランフラグメント：グローバルプランからエージェントごとにサブゴールに分割した。また，ここではサブゴールに対してプランフラグメントを一つ対応させた。

この通信網の例題では制約が弱く，比較的容易に解が求まる場合と制約が強過ぎてすべてのグローバルゴールが満足できない場合の二通りが作成できる。ここでの実験に使用した問題の数は次のとおりである。

- グローバルゴールがすべて満足できるもの—131個。
- すべてのグローバルゴールは満足できないもの—24個。

5.3 実験内容

上記方針に基づいてマルチステージネゴシエーションの枠組での問題を作成し，次に作成された問題を分散制約充足問題へと変換し，各々の問題について処理時間を測定した。処理時間はステージ数で測定した。ここでいうステージとは1ステージにおいて各エージェントが一つのメッセージを処理することができるものである。したがってエージェント内における処理に比べ通信に時間がかかるような場合，ステージ

数を実際の処理時間の第一次近似としてとらえることができる。

非同期バックトラックのアルゴリズムは文献15)に基づいて実装した。そこでは一つのエージェントに対し，一つの変数が対応していることを前提としており，本論文のように一つのエージェントに複数の変数が存在する場合に対処するために3.2節で述べたように変数にidを割り振った。また，次のような実装を行った。

- 一つのエージェント内に複数の変数が存在する場合，自エージェント内の変数に対するメッセージは仮想的に自分自身に対してメッセージを送るよう実装したが，測定においてはステージ数においては仮想的なメッセージの処理に対しても単位ステージの処理がかかるとした。これはステージ数が実際の処理時間の近似としてとらえているからである。
- 非同期バックトラックにおいては変数の全順序をつける必要があるが，ここでは対応するグローバルゴールでまず順序付けをし，その中では対応するサブゴールのグローバルプラン（パスの候補）に従って順序付けを行った。これにより，同じグローバルゴールに対応する変数は全順序をつけた中で固まって出現することになるので，あるグローバルゴールのパスの探索中は他のグローバルゴールに対応する変数にバックトラックする可能

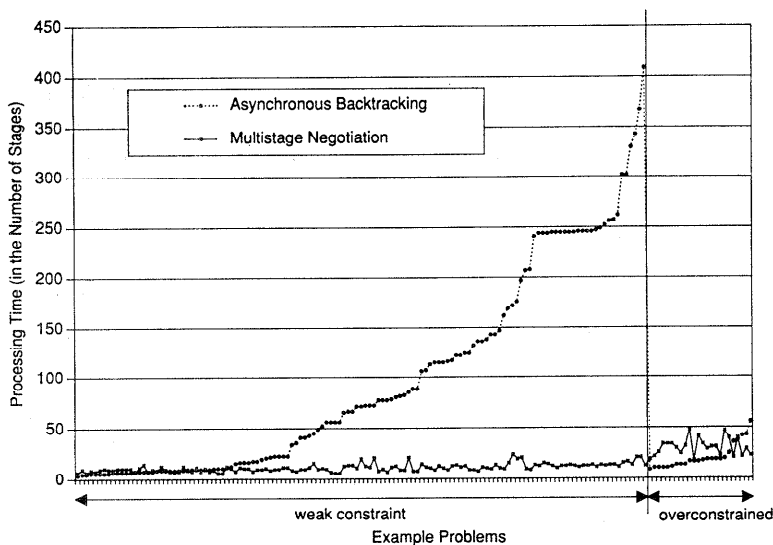


図4 処理時間(ステージ数)

Fig. 4 Processing time (in the number of stages).

性が低くなると考えられる。

5.4 実験結果

図4に実験結果を示す。横軸には例題をとり、縦軸には各々の問題に対するステージ数をとった。横軸は、まず制約が弱くグローバルゴールがすべて満足できるものについて非同期バックトラックにおけるステージ数の少ない順に問題を並べ、次に過制約の問題について非同期バックトラックにおけるステージ数の少ない順に問題を並べた。過制約の問題についてはマルチステージネゴシエーションでは過制約であることを認識する協調探索フェーズまでのステージ数をプロットしている。ただし、フェーズ自体はすべてのエージェントで一斉に切り替わるものではないので、ここではどれか一つのエージェントが次のフェーズに移行した時をもってフェーズが切り替わったとしている。

5.4.1 すべての解が求まる問題

最初の種類の問題に関してはマルチステージネゴシエーションではすべて非同期探索フェーズで解を見つけることが可能であった。非同期バックトラックではバックトラック無しに（すなわち、NoGood のメッセージが流されることなく）解が見つかるものとバックトラックを含むものが含まれている。

このグラフからわかるようにマルチステージネゴシエーションではステージ数にそれほど大きなばらつきがないものの、非同期バックトラックにおいては問題によってステージ数が大きく異なる結果が得られた。

5.4.2 過制約の問題

図5は図4の過制約の問題の部分を見やすくするために拡大したものである。ここでは図4と違ってマルチステージネゴシエーションに関してはフェーズごとのステージ数を示してある。また、過制約解消フェーズもあわせてプロットしている。過制約の問題においてはマルチステージネゴシエーションの方が非同期バックトラックに比較して多くの場合、過制約であることを認識するのにステージ数が多くなっていることがわかる。

5.5 考察

以上の実験結果より、3フェーズプロトコルに関して次のようなことがいえる。

性能上の比較（非同期探索フェーズの効果）：

非同期バックトラックでは基本的にはバックトラックはより高位の変数に対して行われる。バックトラックを行う前にはそれより低位の変数の中ですべての組合せを尽くすことが必要となる。したがって、たまたま、高位の変数において後で制約違反を引き起こすような値の割当をした場合、バックトラックを起こすために大きな処理時間がかかってしまう。

一方、マルチステージネゴシエーションの非同期探索フェーズにおけるバックトラックに相当する処理では非同期バックトラックにおける id を意識しているわけではなく、競合が起こった時にローカルに競合を回避するように別のサブゴールが選択される。したがって非同期バックトラックのように高位の変数に対してバックトラックするために低位の変数であらゆる組合せを試すことはしない。非同期バックトラックにおいて高位の変数で制約違反を起こすような値に設定されてしまうとなかなか制約違反の原因となる変数に対しバックトラックがかからず、問題によって解をみつける時間に極めて大きなばらつきがでてしまったと解釈できる。したがって制約が比較的弱くすべての解が存在する時にはマルチステージネゴシエーションにおける非同期探索フェーズが効果を発揮することがわかる。

過制約の問題における機能と性能のトレードオフ：

非同期バックトラックにおいてはどこかのエージェントで空の NoGood が生成された段階で制約が強過ぎることが認識される。したがって、すべての制約違

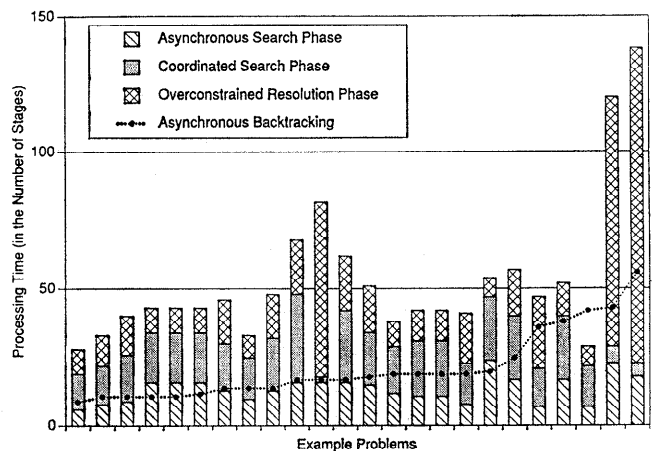


図5 処理時間（ステージ数）—過制約の場合
Fig. 5 Processing time (in the number of stages)—when overconstrained.

反の組を挙げるわけではない。一方、マルチステージネゴシエーションではすべての競合関係を調べるために協調探索フェーズにおいてはすべての組合せをチェックしたかを調べるために *acknowledge* を使ったプロトコルを規定している²⁾。そのため、最終的に過制約であると認識するためにはすべての競合関係を調べ尽くすことが必要になる。さらに *acknowledge* などのプロトコルのオーバーヘッドがあるため、過制約であることを認識するのに時間がかかっていると考えられる。

しかし、一方、マルチステージネゴシエーションにおいて得られる NoGood Goal Set にはグローバルゴール間の競合関係の情報が入っているののでどのグローバルゴールをあきらめれば残りのグローバルゴールに関しては解くことができるかが NoGood Goal Set を調べればすぐわかる。(分散制約充足問題においてはグローバルゴールという概念が無いので、どのグローバルゴールをあきらめたらよいかは自明ではない。) さらに、その時点において各グローバルプラン、サブゴールのレベルにおける競合に関しても情報が得られているのでどのグローバルゴールについて解くかを決定した段階でバックトラック無しに解を求めることが可能になっている。

したがって、マルチステージネゴシエーションにおいては過制的状態を認識するのに時間がかかるものの、いったん認識した後はどのグローバルゴールをあきらめたらよいかなどの情報が得られているという特徴がある。

6. おわりに

ここではマルチステージネゴシエーションの枠組を分散制約充足問題として見た時、どのようにマッピングされるかを明らかにし、さらに、実際の通信網のデータをもとにした例題を使って、マルチステージネゴシエーションと分散制約充足問題の基本的な解法である非同期バックトラックとの性能の比較を行った。その結果、すべてのグローバルゴールの解が求まるような比較的容易な問題に対してはマルチステージネゴシエーションにおける非同期探索フェーズが効果を発揮し、非同期バックトラックに比べ、少ない処理時間(ステージ数)で解を求めることができることがわかった。また、制約が強過ぎてすべてのグローバルゴールの解が求められないような問題に対しては、マルチステージネゴシエーションがすべての組合せの情報を

集めていることになり、非同期バックトラックに比べ、多くの場合時間がかかる。しかし、非同期バックトラックで求められる NoGood の情報だけでは必ずしも、どのゴールをあきらめれば、残りのゴールに関して解くことができるかはわからない。

今回の評価実験では過制約の問題とそうでない問題に場合について主に考察したが、問題の複雑さの処理時間に与える影響、また非同期バックトラック以外のアルゴリズムとの比較は今後の課題である。

また、非同期バックトラックでは変数の順序が解の探索に重要な意味を持ってくる。これについてはヒューリスティクスを導入して変数の並び替えを行った時の効果が報告されている⁹⁾。今回の通信網に基づいた例題においても変数の順序を変えることによりスピードアップが図れることが予想される。実際にどのように変数の順序を変えるか、さらにどのようにマルチステージネゴシエーションにおける非同期探索フェーズのような探索制御を導入すればよいかは今後の課題である。

さらに、マルチステージネゴシエーションにおいてはバックトラックの順序に関してあまり考慮していなかった(バックトラックに相当する *retry* のメッセージに関してどのサブゴールをあきらめるかは適当に決めている)。これに関して何らかのヒューリスティクスを導入することにより、性能の向上が期待することができよう。

また、今回はマルチステージネゴシエーションのもとの出発点である通信網の例題に基づいて評価実験を行った。マルチステージネゴシエーションをより広い枠組で考えようとする、他の応用においてさらに評価を行う必要がある。

謝辞 ネットワーク構成管理データベースを提供していただいた NTT 伝送システム研究所の藤井伸朗主幹研究員、依田育生研究主任に感謝します。また、日頃より御指導いただく NTT コミュニケーション科学研究所の西川清史所長、中野良平グループリーダー、また、草稿に多くのコメントをいただいた同所の石田亨主幹研究員、議論いただいた同所の横尾真主任研究員、西部喜康社員に感謝します。

参 考 文 献

- 1) Collin, Z., Dechter, R. and Katz, S.: On the Feasibility of Distributed Constraint Satisfaction, *AAAI-91*, pp. 318-324 (1991).
- 2) 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).
- 3) Conry, S. E., Meyer, R. A. and Lesser, V. R. : Multistage Negotiation in Distributed Planning, in Bond, A. H. and Gasser, L. eds., *Readings in Distributed Artificial Intelligence*, pp. 367-384, Morgan Kaufmann (1988).
- 4) Davis, R. and Smith, R. G. : Negotiation as a Metaphor for Distributed Problem Solving, *Artificial Intelligence*, Vol. 20, No. 1, pp. 63-109 (1983).
- 5) 桑原和宏 : マルチステージネゴシエーションによる分散資源割当, 1990年代の分散処理シンポジウム, pp. 87-96, 情報処理学会 (1990).
- 6) Kuwabara, K. and Lesser, V. R. : Extended Protocol for Multistage Negotiation, *9th Workshop on Distributed Artificial Intelligence*, pp. 129-161 (1989).
- 7) 桑原和宏, Lesser, V. R. : マルチステージネゴシエーションにおけるゴール間競合の検出, 情報処理学会論文誌, Vol. 32, No. 10, pp. 1269-1280 (1991).
- 8) Mackworth, A. K. : Constraint Satisfaction, in Shapiro, S. C. ed., *Encyclopedia of Artificial Intelligence*, pp. 205-211, John Wiley & Sons (1987).
- 9) Nishibe, Y., Kuwabara, K. and Ishida, T. : Effects of Heuristics in Distributed Constraint Satisfaction : Towards Satisficing Algorithms, *11th International Workshop on Distributed Artificial Intelligence*, pp. 285-302 (1992).
- 10) Smith, R. G. : The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers*, Vol. 29, No. 12, pp. 1104-1113 (1980).
- 11) Smith, R. G. and Davis, R. : Frameworks for Cooperation in Distributed Problem Solving, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1, pp. 61-70 (1981).
- 12) 山口治男, 藤井伸朗, 山中康史, 依田育生 : ネットワーク構成管理データベース, *NTT R & D*, Vol. 38, No. 12, pp. 1509-1518 (1989).
- 13) 依田育生, 山中康史, 藤井伸朗 : 通信網オペレーションシステムにおけるネットワーク構成管理データベース, 電子情報通信学会秋季全国大会, B-254 (1989).
- 14) Yokoo, M., Durfee, E. H., Ishida, T. and Kuwabara, K. : Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving, *12th International Conference on Distributed Computing Systems*, pp. 614-621 (1992).
- 15) 横尾 真, エドモンド H. ダーフィ, 石田 亨, 桑原和宏 : 分散制約充足による分散協調問題解決の定式化とその解法, 電子情報通信学会論文誌 (D-I), Vol. J 75-D-I, No. 8, pp. 704-713 (1992).
- (平成4年9月10日受付)
(平成5年4月8日採録)



桑原 和宏 (正会員)

1959年生。1982年東京大学工学部電気工学科卒業。1984年同大学院電子工学専門課程修士課程修了。同年日本電信電話公社入社。以来知能ベースシステム、分散協調問題解決などの研究に従事。1988年9月より一年間 University of Massachusetts at Amherst 客員研究員。現在, NTT コミュニケーション科学研究主任研究員。