

## 分散協調問題解決の契約に関する考察

須崎 健一 荒屋 真二

(福岡工業大学 工学部 通信工学科)

分散協調問題解決システムとは、多くのプロセッサ（エージェント）に分散化された知識源により協調的に問題解決を行なうシステムである。このようなシステムはいくつか提案されているが、その中で最も重要なものに契約ネットがある。この契約ネットの問題点の一つに、実行中の問題を一つでも抱え込んでいるエージェントは、決して別の入札には応じられないということがある。そこで、本稿では一つのエージェントが他の複数のエージェントと異なる契約を並列的に結べるような枠組みを提案する。この拡張された契約ネットを実現するために、エージェントの負荷量という概念を明示的に導入する。この負荷量から、問題を依頼する際に付ける納期や問い合わせに対する回答予定時刻の算出が可能となる。ここでは、提案方式を従来の契約ネット方式と比較することにより、提案方式の有用性を示す。

### Consideration on Contracts in Distributed Cooperative Problem Solving Systems

Kenichi SUZAKI and Shinji ARAVA

Fukuoka Institute of Technology

3-30-1, Wajirohigashi, Higashi-ku Fukuoka, 811-02, Japan

Distributed cooperative problem solving systems are AI which cooperatively solve problems by using knowledge sources distributed among many processors (called "agents"). Several such systems have been proposed so far, and the most important may be the contract net. The contract net, however, has an imperfection of its own, that an agent which is solving a problem is never able to response to another bid. This paper proposes a framework that each agent can make several different contracts inparallel with other agents. To implement this expanded contract net, the concept of "load" of each agent is explicitly introduced. The load allows calculation of expiration times attached to the problem referred to the agent and response times to inquiries. In this paper, we shall compare the proposed net with the contract net in order to demonstrate it's usefulness.

## 1. まえがき

分散協調問題解決システムとは、多くのプロセス（以下エージェントと呼ぶ）に分散化された知識源により協調的に問題解決を行うシステムである。ここでは、どのエージェントも単独では全体問題を解くことができず、情報の相互通信が不可欠である。分散化とは、単に知識だけでなく制御も地理的あるいは論理的に分散していることを意味する。つまり、大局的共有メモリも大局的制御も存在しておらず、この点が従来の分散処理と異なる点である。もちろん、従来の分散処理の利点である効率性、信頼性、資源の有効利用、拡張性などは、分散協調問題解決システムにも継承される。特に重要な利点は、エージェント故障時の対応能力が優れていること（フェイルソフト）、不確定かつ不完全な情報の存在を許容できることである。また、真に知的なシステムは、複数の協調的サブシステムに分解しなければ実現できないほど膨大な知識を必要とする。一人の人間が効率的に管理できるデータや制御の量には限度があり、複雑な問題を解くには複雑な組織が必要となる（Simon）。現実の全てのシステムは本質的に分散的であり（Hayes-Roth）、知能の認知科学的解明に不可欠な要素である（Nilsson）。分散人工知能は、粒度の細かいコネクショニストから、粒度の大きい分散エキスパートまで多様な形態を取り得るが、本稿で取り上げる分散協調問題解決システムは後者に近いものである。

本稿では、次の五つの大前提をおく。即ち、1)疎結合非同期エージェントからなるネットワークである、2)各エージェントがもつ知識源はかなり異なっている、3)各エージェントはメッセージ交換により相互通信が可能である、4)エージェント間でメモリの共有は行わない、5)高信頼、高効率の低水準通信プロトコルは存在している。これらは、人間組織における問題解決のメタファとしての根回しを基本モデルとした契約ネット方式 [1] と同じ前提である。筆者等が現在開発中のシステムは、この契約ネットの基本的考え方の大部分を踏襲している。契約ネットでは、解決すべき課題をもつエージェントがマネージャとなり、その副課題を解く知識源をもつ適切なコントラクターを根回しにより選択し契約を結ぶ。ここで根回しというのは、情報の評価があくまで局所的に行われ、かつ全てのエージェントが相互選択権を持つことを意味する。全てのエージェントはマネージャにもコントラクターにもなることが可能であり、これは制御階層が状況に応じて動的に変化することを意味している。

契約ネットの大きな問題点の一つは、実行中の課題の一つでも抱えているエージェントは決して入札に応じないということである。故に、解決可能な能力をもつエージェントの全てが何等かの課題を実行

中のときには、常にBUSYが返って来るので、ある時間間隔毎に再依頼のメッセージが何度も送信される。ネット全体の負荷が大きい時には、このような無駄なメッセージが飛び交い、通信チャンネルの飽和を招いてしまう。この問題に対処するために、契約ネットでは、遊んでいるエージェント側からも課題を求めてメッセージを送信できる機能を追加し、両者をネット負荷に応じて切り替える方法を提案している [1]。ネット負荷が小さい時には新たな課題の発生が重要事象であり、負荷が大きい時には暇なエージェントの発生が重要事象となるからである。この志願メッセージの導入は通信のオーバーヘッドの軽減に貢献するが、具体的な切替方法やその効果についてはまだ明らかになっていない [1]。また、この切替をうまく行ったとしても、一つのエージェントは一つの課題きり引き受けることができないため、ネットの負荷が増大すると依頼すべき課題を抱えたエージェントの数が増加する。

そこで本稿では、一つのエージェントが他の複数エージェントと異なる契約を並列的に結べるような枠組を提案する。この拡張された契約ネットを実現するためには、各エージェントの負荷量という概念を明示的に導入してやる必要がある。従来の契約ネットでは、エージェントの状態として、①BUSY（課題実行中）、②INELIGIBLE（暇であるが、その課題を解く能力がない）、③LOW-RANKING（暇で能力もあるが、その課題に対する優先度が低い）の三つだけを考え、負荷量までは考慮に入れていない。この負荷量を抜きにしては、納期の設定や納期内に結果を出せるかどうかの判断は不可能である。契約ネットにおいても納期を扱っているが、その設定方法やそれがどのように制御に反映されるかについては触れられていないが、その理由はここにある。依頼メッセージと志願メッセージの切替をうまくやるためにも負荷量は大きな役割を果たす。人間組織において、一人の人間が複数の課題を抱えている状態は自然であり、処理すべき課題がなくなってから次の課題を探す方法には無駄時間が発生する。各エージェントの負荷量を考慮してはじめて負荷バランスも含めた焦点制御が可能となる。

次章では、各エージェントの負荷量を導入することで、契約を並列的に結べる根拠を述べる。3章では、提案する方式のエージェントの各機能を述べて、契約ネット [1] では考察していない納期設定方法、回答予定時刻、およびコントラクターを選定する方法等について考察する。4章では、提案方式と契約ネット方式を比較考察する。5章では、本稿をまとめると共に、今後の課題について触れる。

## 2. 拡張契約ネット

契約ネットにおいて、マネージャは原問題をいく

つかの副問題に分割し、依頼すべきものは同報で問い合わせを行い、入札に応じたものから一人（コントラクター）と契約を結び、正式に依頼をしている。本稿でも同じく依頼すべき副問題毎に1対1の契約を結ぶ方式を採用する。ここで扱う各エージェントのアーキテクチャは、文献[3]と同じであり、フロントエンド、問題計画部および問題解決部の各プロセスから構成されている。ここで扱う各エージェントは、すべて対等であり、各エージェントの役割および関係は動的に変更できるものとする。

契約ネットの欠点の一つは、コントラクターの立場で問題を解いている間は、別問題の問い合わせに対してbusy信号を戻し、二重の契約に応じないということである。したがって、入札に応じられるものは、暇で、かつ、マネージャがつけた納期内に結果回答できるエージェントだけになる。一方、マネージャの立場で問題依頼の問い合わせをしたエージェントは、コントラクターの選定を予想性能だけで判定して契約を結んでいる。正式依頼した後、マネージャは自分で解くべき副問題を解いて、コントラクターからの結果回答を受けて、原問題の結果合成を図っている。そのため、マネージャは既にコントラクターと契約を結んでいる間は、別問題を抱え込むこともなく、また別問題の問い合わせもできないということになる。つまり、各エージェントはマネージャにもコントラクターにもなりえるが、一つでも契約を結んでいる間は、同時に両方の立場にはなりえないのである。したがって、契約ネットでは、マネージャとコントラクター間では、単一の問題しか存在しなく、複数の問題を並列的に処理することができない。

そこで、一つのエージェントが他の複数のエージェントと異なる契約を並列的に結べるような契約方式を提案する。つまり、エージェントがすでに問題を処理中であっても、マネージャ、またはコントラクターの立場になることができる。ここでは、この提案方式を拡張契約ネットと呼び、従来の契約ネットの機能の他に、次のようなものを追加する。

①各エージェントは、依頼すべき問題が発生すれば、依頼したいエージェントに問題の内容と負荷量から計算した納期を付けて問い合わせを行なう。

②各エージェントは、現在、問題を処理中であっても、別問題の問い合わせを受けたときは、その問題の結果が回答できるまでの予定時間（以下、回答予定時刻と呼ぶ）と予想性能、および現在の負荷量を付けて、必ずある一定時間後に返答する。

③各エージェントは、正式依頼を受けた問題をスケジューラに先着順に格納するが、もし、納期切れが間近な問題があれば、スケジュール変更して先に処理できるようにする。

④各エージェントは、コントラクターの選定を行なう際に、予想性能と回答予定時刻を考慮に入れる。

⑤各エージェントは、故障回復した際は、自分宛に問い合わせをしたマネージャに回復した旨のメッセージを送信する。

各エージェントは、上記の機能を持つことで、契約を並列的に結べたり、また故障回復後の対応処置がとれるなど契約機能が拡張されたことになる。ここで、問い合わせを受けた問題の回答予定時刻の算出と自分が依頼するときに付ける納期の設定は、自分が抱え込んでいる負荷量から算出することを述べた。そこで、負荷量とは、①現在処理中の問題を解決するための時間と、②スケジューラに待機中の問題をすべて解決し、回答を完了するまでの時間の和をいう。即ち、負荷量とはエージェントが現在抱え込んでいる各問題が解決し空になるまでの予定時間をいう。ここでは、①の負荷量を実行負荷量と呼び、②の負荷量を待機負荷量と呼ぶことにする。実行中の問題の処理の中断は禁止し、この負荷がなくならない限り、待機中の問題は処理できない。

以下、エージェントの諸機能と拡張契約ネットの評価について述べる。

### 3. エージェントの諸機能

#### 3.1 回答予定時刻の算出

回答予定時刻の算出は、各エージェントが抱え込んでいる負荷量、つまり実行負荷量と待機負荷量から算出する。問い合わせを受けた問題が自分で解けて、スケジューラに先着順に格納してもマネージャが付けた納期まで余裕があれば、その予定時間を回答予定時刻とする。しかし、納期切れになるようであれば、スケジュールの変更を試みる。スケジュールの変更は、納期切れの問題を先に処理できるようにする。いま仮に、スケジューラの中に待機している問題の納期に余裕があり、問い合わせを受けた問題の納期がその余裕より短ければ、その問題を先に処理するように変更する。したがって、回答予定時刻の算出は、現在の実行負荷量とスケジュール変更から定まる問題の解決終了予定時間の和で求まる。もし、スケジュールを変更しても、マネージャが付けた納期を越えるなら受諾は無理であるが、算出した回答予定時刻を返答する。また、解けない問題であれば、無限大の回答予定時刻を返答する。これらの返答に対しては、マネージャの方で、自分の負荷量の各問題をスケジュール変更して、納期の変更が可能か否かを判断して、正式に依頼すべきか否かを判断すべきものである。

#### 3.2 納期の設定

納期には、システム外からエージェントに依頼するときに付ける納期と、そのエージェントがいくつ

かの副問題に分割し、マネージャの立場で問題を依頼するときには付ける納期がある。システム外からの納期は、受諾したエージェントが回答できるまでに充分余裕があるものとする。ここでは、マネージャの立場で依頼する際に付ける納期について考察する。

各エージェントはマネージャでもコントラクターにもなりえるため、納期には自分がコントラクターの立場で受けた納期( $T1$ )と、マネージャの立場でコントラクターに付ける納期( $T2$ )がある。 $T2$ の設定は、 $T1 > T2$ の関係になるようにする。この $T2$ の値は、自分が依頼を受けた際に返答した回答予定時刻よりは、必ず短くなければならない。したがって、 $T2$ の値は回答予定時刻から結果の授受までに要する送・受信時間を引いた残りとなる。この $T2$ がコントラクターでは $T1$ となる。この $T2$ の納期が付いた問題依頼の問い合わせを受けたエージェントは、前章で述べた方法で、回答予定時刻を返答する。

### 3.3 コントラクターの選定

マネージャは問い合わせに対する返答の中から、正式に依頼すべきコントラクターを選定する。従来の契約ネットでは、問い合わせの返答は納期以内に結果回答が可能なエージェントに限られていた。そのため、コントラクターの選定は予想性能だけで行っていた。一方、拡張契約ネットでは、負荷量を明示的に導入したため、問い合わせの返答の中に、予想性能と回答予定時刻が含まれている。そのため、コントラクターの選定は返答内容によって次のいずれかの方法で行なう。

①回答予定時刻が、納期( $T2$ )内のもので最高の予想性能を示したエージェントを選定する。

②もし、すべての回答予定時刻が納期( $T2$ )外であれば、自分がコントラクターの立場で受けた納期( $T1$ )を越えない範囲でスケジュールを変更して、新たな納期( $T2$ )を定め、その納期内に該当する中で最高の予想性能を示したエージェントを選定する。ここで、②でスケジュール変更をしてもすべての回答予定時刻が納期外であれば、その問題は自分が受けた納期( $T1$ )内では解けないことになる。

## 4. 拡張契約ネットの評価

拡張契約ネットを、問い合わせ回数、各問題のスケジュール変更による納期切れの回避、およびエージェント故障による問題処理の影響の点から、従来の契約ネットと比較する。

### 4.1 問い合わせ回数

分散協調問題解決システムの主要な評価指標の一

つに、分散された問題を解き終わるまでに用いられたメッセージ総数がある。そこで、契約ネットと拡張契約ネットにおいて、ある問題が契約されるまでの問い合わせ回数を一例にとり、両者の違いについて考察する。

ここで、問い合わせ回数の評価を簡単化するために、次の五つの仮定を設定する。(1)システム上の各エージェントは共通した時間を持ち、エージェントでの問題処理は1単位時間( $t$ )の整数倍で表わすことにする。(2)エージェント間でメッセージを交換する際の伝送時間と、システム外から問題を依頼する際の問い合わせ回数は無視する。(3)エージェントが、システム外から問題の依頼を受けた後、その問題の結果回答までに要する時間を、 $t_d$ 、 $t_a$ 、 $t_p$ および $t_c$ の四つに分けて考える。 $t_d$ は、依頼を受けた問題をいくつかの副問題に分割し、問題解決の計画を立てるのに要する時間(1単位時間)。 $t_a$ は、依頼すべき副問題を、各エージェントに問い合わせを行い、候補者を選択し、コントラクターに正式依頼するまでに要する時間(1単位時間で、これを1回の問い合わせ数とする)。 $t_p$ は、(副)問題の解決に要する時間(数単位時間)。および、 $t_c$ は、すべての副問題の解が得られた後に、結果を合成するのに要する時間である(1単位時間)。(4)コントラクターは契約した問題を $t_d$ で分割したのち、 $t_p$ で解決しマネージャに回答するものとする。(5)システム外とマネージャからの問題依頼が同時刻にあった場合は、システム外からの依頼を優先する。

図1は、契約ネットにおいて、3個の各エージェントに、ある一定時間(5 $t$ )毎にシステム外から問題の依頼を行い、その問題が契約され、解決されるまでの過程を単位時間の経過で表わしたものである。図の $Pa1$ とは原問題を、 $Pa1'$ と $Pa1''$ は $Pa1$ の各副問題を表わす。以下、同様な形式で問題または副問題を表わすこととする。ここで、副問題はエージェント数だけ分割するものとする。図1から分かるように、契約ネットにおけるコントラクターは、実行中の問題があれば、別の問題の問い合わせがあっても、契約を結べないため $busy$ を返す。そのためマネージャは、 $busy$ 以外の回答が得られるまで単位時間毎に問い合わせを行なう。契約ネットのエージェントはマネージャにもコントラクターにもなりうるが、一つの問題を処理中には同時にはなりえない。即ち、一つの問題を処理中には別の問題の契約は結べないため、別の問題の依頼もすることができない。この制約は、仮に両者間で互いに依頼すべき副問題があれば、片方が中断しない限り先に進めない。ここで、中断については、コントラクターは、マネージャと契約した問題のみ中断をするのであって、別のマネージャから別問題の依頼があったからといって、現在処理中の問題を中

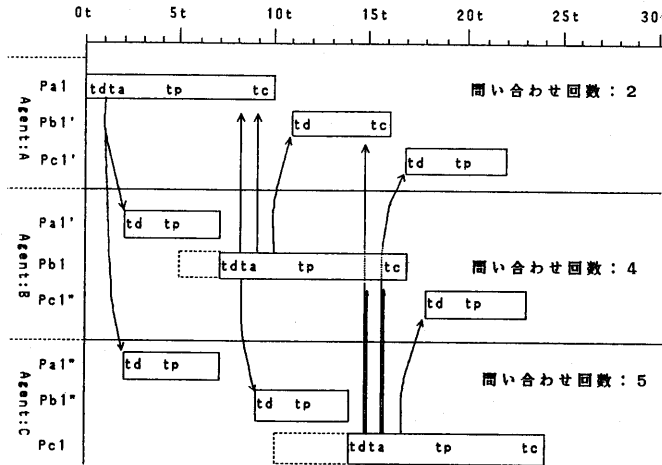


図1. 契約ネットにおける問い合わせ回数

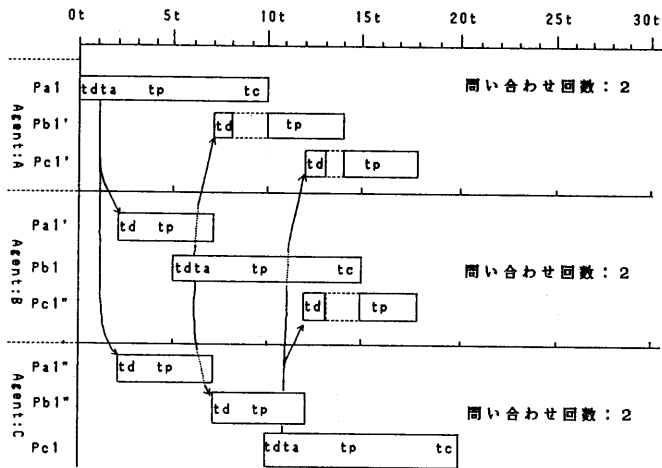


図2. 拡張契約ネットにおける問い合わせ回数

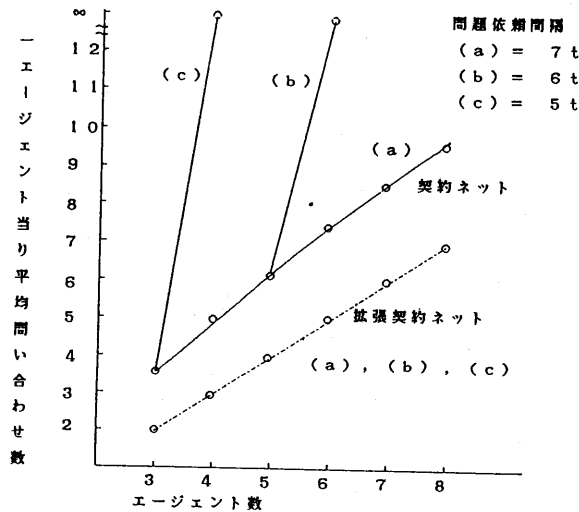


図3. 1エージェントの平均問い合わせ回数

断することはしない。したがって、両者がマネージャの立場であり続けるとデッドロックに陥ってしまう。図1において、エージェント数を4台 (Agent: A ~ D) にし、図1と同じ条件で問題の依頼を与える続けると、15t経過後にエージェントCとDの両者の間で互いに相手に依頼すべき副問題が発生してしまい、デッドロックに陥ってしまう。

一方、拡張契約ネットでは、各エージェントは同時にマネージャにもコントラクターにもなりえて、契約を並列的に結ぶことができる。図2は、拡張契約ネットにおいて、図1と同じ時刻に同じ問題を依頼したときの各問題の処理経過を示したものである。図2は図1に比べて、問い合わせ回数も少なく、問題解決も早く終わっている。図3はエージェントの数を増し、各エージェントに図1および2と同じ条件の問題を依頼することにして、その時間間隔を7t、6t、および5t (図1、2と同じ依頼間隔) としたときの1エージェントあたりの平均問い合わせ回数を表わしたものである。図3より、契約ネットは依頼間隔を7t以上にすればデッドロックに陥らず問題処理が行えるが、その問い合わせ回数はコントラクターが現在処理中の問題が終了するまでの時間に比例して増えてくる。一方の拡張契約ネットでは、依頼間隔に関係なく自分を除いたエージェント数 (副問題を自分を除いたエージェントに分配する) だけに問い合わせをすればよく、契約ネットに比べて平均問い合わせ数も少なく、デッドロックに陥ることもない。

#### 4. 2 問題のスケジュール変更

契約ネットでは複数の問題の依頼が同時にあっても、契約は一つだけしか結べないので、最初の問題が受理可能で契約してしまえば、後の問題は無視されてしまう。したがって、先着順で前もって契約を結んでおく機能もなく、スケジュールを立てることもできない。図4は、契約ネットにおいて、システム外からエージェントAに0t, 3t, 6tの各時刻に問題Pa1 (納期10t), Pa2 (納期25t), およびPa3 (納期20t) を、またエージェントBに、10tの時刻に問題Pb1 (納期25t) を依頼し

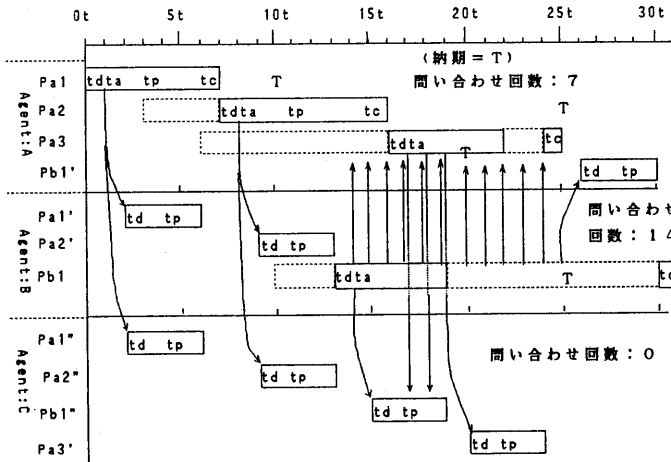


図4. 契約ネット (スケジュール変更なし)

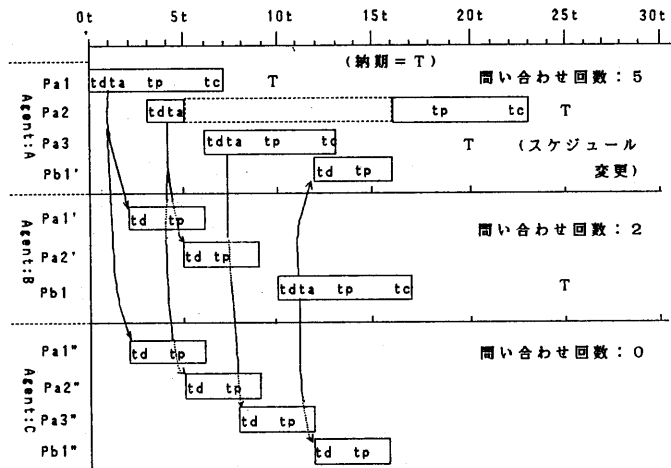


図5. 拡張契約ネット (スケジュール変更あり)

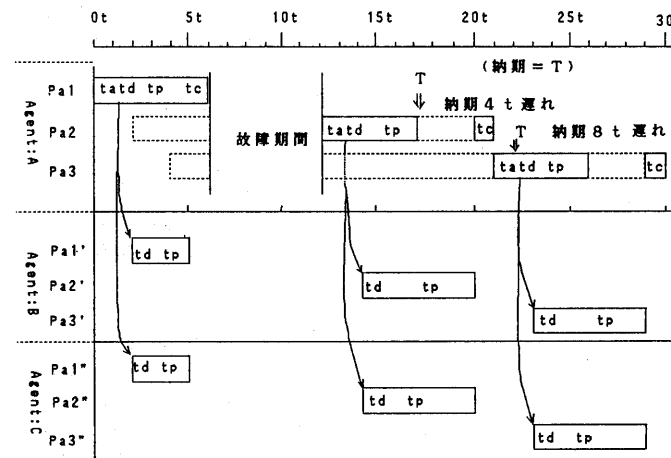


図6. 契約ネットにおけるマネージャの故障

たときの各問題の処理経過を示したものである。

契約ネットでは、図4からわかるように、スケジュール変更ができないため、一つの問題を解き終えてから次の問題に着手している。そのため、問題Pa3とPb1ともに納期切れになってしまう。

一方、拡張契約ネットでは、複数の問題を並列的に契約できるので、ある問題を処理中でも、新たな契約が結べる。いま、ある問題を処理中に、新しい問題の依頼があったと仮定し、その問題が先着順で契約を結んでも、納期までに終わる予定が立てば最後に契約した問題としてスケジューラに追加する。しかし、先着順では納期切れになるのであれば、スケジュールの変更を行なう。図5は問題の依頼を図4と同じ条件で与えた場合において、スケジュールの変更を行なった例である。図5から、エージェントAは納期の長い問題Pa2を後にずらすことで、すべての問題を納期内で終了させている。同様に、エージェントBも問題Pb1を納期内で終了させている。

#### 4.3 エージェントの故障

エージェントの故障とは、受信機能を除いて、各プロセス（フロント・エンド、問題計画部、および問題解決部の通常処理）の機能が一時停止している状態をいう。また、エージェントの故障はリンクに何ら影響を与えず、リンク自体の故障はないものとする。エージェントの故障検出は、送信メッセージを飛ばしたエージェントが、ある一定時間経過しても相手のエージェントから返答がないとき、そのエージェントは故障していると判断する。エージェントの故障回復ができた場合は、故障したエージェントで実行されていたプロセスを故障発生直前の状態に戻ることができるものとする。また、故障中に送信されてきたメッセージの履歴も回復することができるものとする。故障回復したエージェントは、故障中に送信してきたエージェントに対して、回復した旨のメッセージを送信する。回復メッセージを受信したエージェントは、故障したエージェントが回復したことを知り、通信し合うことができる。次に、マネージャもしくはコントラクターがある時間だけ故障した場合の問題処理に与え

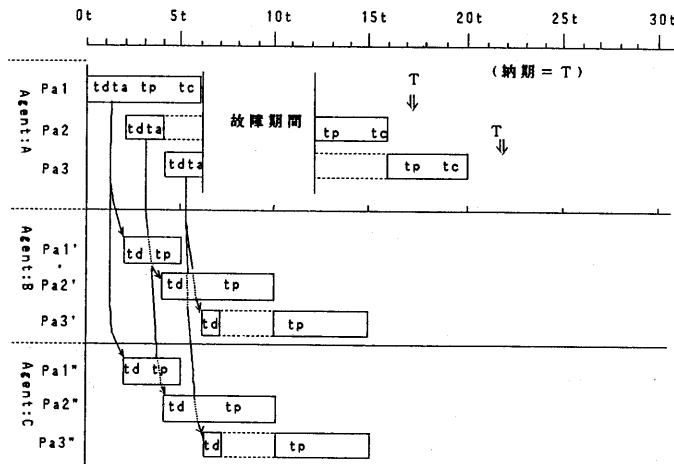


図7. 拡張契約ネットにおけるマネージャの故障

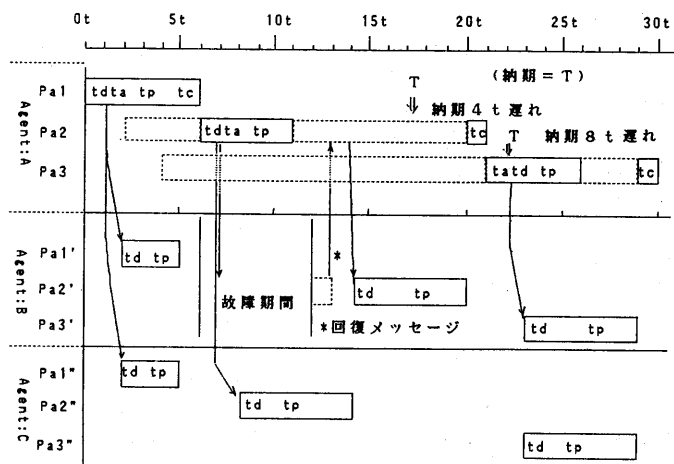


図8. 契約ネットにおけるコントラクターの故障

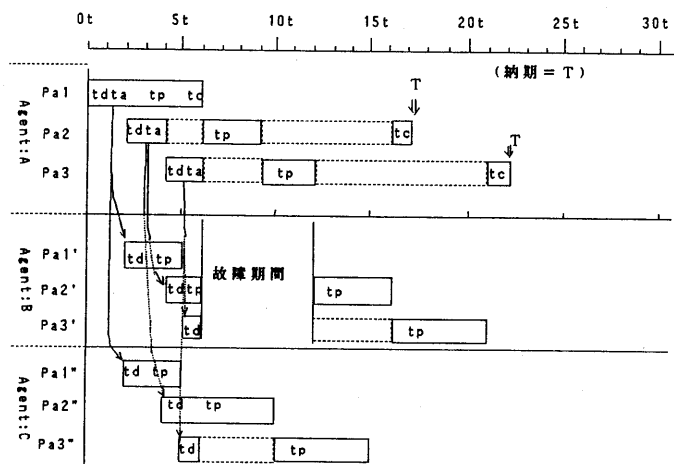


図9. 拡張契約ネットにおけるコントラクターの故障

る影響を、契約ネットと拡張契約ネットの両方の違いについて考察する。

#### 4. 3. 1 マネージャの故障

図6は契約ネットにおいて、エージェントAに0t, 2t, および4tの時刻に問題Pa1, Pa2およびPa3を依頼したとき、エージェントAだけが6tで故障し12tで回復した場合の各問題の処理の経過を示したものである。図6から、マネージャが単独で故障した場合は、すべてのコントラクターにも、その影響がでている。マネージャの故障による遅れは、原問題の着手まで遅れてしまうため、依頼すべき副問題の問い合わせまで遅れてしまう。したがって、故障による問題解決の遅れが、その問題に関与したエージェントすべてに影響してしまう。

図7は拡張契約ネットにおいて、図6と同じ時刻に同じ問題を依頼した場合の各問題の処理経過を示したものである。エージェントAは契約直後にtd, taと進み、依頼すべき副問題を2単位時間後に依頼している。これは、いつまでも依頼すべき副問題を手元に抱え込まず、契約ネットに比べて早く依頼ができることを示す。図7から、エージェントAの故障は副問題を依頼した後に発生しているため、故障期間中に他のエージェントが副問題を解き終わっている。したがって、回復後は、自分が解くべき副問題を処理するだけでよく、故障による遅れを最小限に抑えることができる。

#### 4. 3. 2 コントラクターの故障

コントラクターは、故障期間中でも自分宛に送られてきたすべてのメッセージをフロント・エンドの受信待行列に格納している。そのため、故障回復時には待行列のメッセージを読み取り、自分への問い合わせの有無を知ることができる。もし、問題依頼の問い合わせがあれば、マネージャに故障回復したことを通知する。その後、故障以前に処理していた問題があれば、故障の発生前に立ち戻り処理を続行する。

図8は、図6と同じ条件で問題を依頼したときに、エージェントBに故障が発生した場合についての各問題の処理経過

を示したものである。また、図9は、図7と上記と同様な関係を表わしたものである。図8と図9からわかるように、拡張契約ネットと契約ネットともに同じ条件（問題の依頼、故障期間、および故障回復時の処置等）であれば、拡張契約ネットの方が契約ネットに比べて、早く依頼されるべき副問題を契約しているため、問題解決に着手する時期がその分早くなる。これは、拡張契約ネットの方が契約ネットに比べて、納期内に結果回答できる可能性を示している。

## 5. まとめ

一つのエージェントが、他の複数のエージェントと異なる契約を並列的に結べるような拡張契約ネットを提案した。拡張契約ネットと従来の契約ネットを、問い合わせ回数、各問題の納期遅れ、およびエージェント故障時の柔軟性の観点から比較考察した。その結果、拡張契約ネットの方が、次のような点で優れていることが明らかになった。

①提案した契約方式では、問い合わせは1回でよく、契約ネットのようにbusy信号を受ける度毎に何回も繰り返さなくてすむ。また、ある問題の契約中でも依頼または受諾エージェントの立場になれるため、デッドロックに陥ることもない。

②契約した問題は、スケジューラに先着順に格納されていくが、スケジュールの変更により納期遅れを最小化できる。

③エージェントの故障が、問題処理に与える影響については、依頼を受けた問題、時期等が同じ条件であれば、拡張契約ネットの方が契約ネットに比べて、早く問題を分配して依頼するので、故障の発生

が依頼した直後であれば、故障による遅れを自分だけにとどめ、故障の被害を最小限に抑えることができる。

今後、筆者らは、本稿で提案した拡張契約ネットを、より一般的な条件のもとで実験的に評価してゆく予定である。

## [参考文献]

- [1] Smith, R.G.: The Contract Net Protocol / High Level Communication and Control in Distributed Problem Solver, IEEE Trans. Comput., Vol.C-29, Dec.(1980).
- [2] Smith, R.G. and Davis, R.: Frameworks for Cooperation in Distributed Problem Solving, IEEE Trans. Syst., Man, Cybernet., Vol.SMC-11, No.1(1981).
- [3] Yang, J.D., Huhns, M.N. and Stephen, L.M.: An Architecture for Control and Communication in Distributed Artificial Intelligence Systems, IEEE Trans. Syst., Man, Cybernet., Vol.SMC-15, No.3(1985).
- [4] Lesser, V.R. and Corkill, D.D.: Functionally Accurate, Cooperative Distributed Systems, IEEE Trans. Syst., Man, Cybernet., Vol. SMC-11, No.1(1981).
- [5] Pearl, J., Leal, A. and Saleh, J.: GODDESS/ A Goal-Directed Decision Structuring System, IEEE Trans. Pattern Anal. Machine Intel., Vol.PAMI-4, No.3(1982).
- [6] Decker, K.S.: Distributed Problem-solving Techniques: A Survey, IEEE Trans. Syst. Man, Cybernet., Vol.SMC-17, No.5(1987).