

# ネットワーク資源割当問題向き協調問題解決手法

和 氣 弘 明<sup>†</sup> 村 山 隆 彦<sup>†</sup> 服 部 文 夫<sup>†</sup>

ネットワーク資源を経路設定などの要求に割り当てる資源割当問題において、要求をエージェント（自律的処理主体）に分配し、複数エージェント間の協調により問題解決を行う協調問題解決手法を提案する。各エージェントは、分配された要求について自律的に資源割当を行う。エージェント間で資源競合（同一資源への重複割当）が生じた場合は、エージェントは協調動作としてネゴシエーションを行い、競合を解消する。ネゴシエーション戦略には、処理時間の低減と解の質向上を狙った戦略を用いた。提案手法をネットワークの再編成を想定した資源割当問題に適用し、シミュレーションにより集中手法と比較した。その結果、ネットワーク規模の大きい領域では集中手法の約1/2の処理時間で質の高い近似解を得ることができ、提案手法の有効性を確認した。

## Cooperative Problem Solving Suited for Network Resource Allocation

HIROAKI WAKI,<sup>†</sup> TAKAHIKO MURAYAMA<sup>†</sup> and FUMIO HATTORI<sup>†</sup>

This paper proposes a cooperative problem solving method suited for the network resource allocation problem that allocates network resources to each demand. The method assigns demands to agents, and each agent allocates its own demands autonomously. If resource conflict occurs, agents involved in the conflict negotiate with each other, so that resource conflict is eliminated autonomously. We applied the proposed method to the network resource allocation problem, and compared it with the centralized method through simulations. The simulation's results show the effectiveness of the proposal method in large-scale networks.

### 1. はじめに

ネットワーク資源を経路設定などの要求に割り当てる資源割当問題を考える。このような問題は一般にNP完全問題となるため、対象とするネットワークが大規模であったり、要求数が多いなど、規模の大きな問題に対しては、実用的な時間で最適解を求めることが困難であることが知られている。そのため、必ずしも最適解が得られなくても、実用上問題のない近似解を妥当な時間で求めることのできる集中手法<sup>1),2)</sup>が提案されている。

しかし、ネットワーク資源の効率的運用を目指した再編成（資源割当の見直し）が頻繁に行われるなど、ネットワーク資源や要求が動的に変化するような状況では、これらを一元的に管理する集中手法では迅速な対応は困難となる。

このような大規模・動的な問題の場合、問題を分割してエージェント（自律的処理主体）に割り当て、エー

ジェント間の協調により問題解決を行う協調問題解決手法<sup>3),4)</sup>が有効であると考えられる。協調問題解決手法では何をエージェントに割り当てるかが課題の1つであり、従来は主にネットワーク資源という側面に注目して、部分ネットワークを各エージェントに割り当てる手法<sup>5),6)</sup>が用いられてきた。しかし、必ずしも都合良くネットワークを分割できるとは限らず、分割の仕方によっては特定のエージェントに要求が集中するなどの問題が生じる。

一方、要求という側面に注目してみると、時系列的な要求の発生や要求の変更などの動的な変化は、ネットワーク資源よりも要求ごとに発生しやすく、コスト最小化や通信品質最適化といった要求ごとに異なる最適化目的（制約）を持つなどの特徴が考えられる。この場合、個々の要求をエージェントに割り当てる方が管理が簡単となり、処理を行いやすい。そこで、我々は個々の要求を各エージェントに割り当て、複数のエージェント間の協調により問題解決を行う協調問題解決手法を提案する。

本稿では、まずネットワークにおける資源割当問題の定式化を行い、その解決手法を提案する。さらに、

<sup>†</sup> NTT 情報通信研究所

NTT Information and Communication Systems Labs.

提案手法をネットワークの再編成を想定した資源割当問題に適用し、シミュレーションにより集中手法と比較/評価を行う。

## 2. ネットワーク資源割当問題

本稿で扱うネットワークにおける資源割当問題を図1に示す。ネットワークはノードと容量付きのリンクからなる無向グラフとする。このネットワークにおいて、2点間に一定容量の経路を設定する経路設定要求を考える。ただし、経路設定時に、経路を構成するリンクごとに使用量に応じたコスト（使用コスト）がかかるものとする。

ここに、ネットワーク資源割当問題を、各リンクの容量を超えないように、かつ使用コストの総和が最小となるように、ネットワークのリンクを複数の要求に割り当てる問題と定義し、以下のように定式化する。

$n_i$ :  $i$  番目のノード,  $i = 1, 2, \dots, M$

$l_{i,j}$ :  $n_i, n_j$  間のリンク,  $l_{i,j} = l_{j,i}, i \neq j$

$lc_{i,j}$ :  $l_{i,j}$  のリンク容量,  $lc_{i,j}$  は正整数

$uc_{i,j}$ :  $l_{i,j}$  における単位使用コスト

$r_k$ :  $k$  番目の要求,  $k = 1, 2, \dots, K$

$sn_k$ :  $r_k$  の始点ノード

$en_k$ :  $r_k$  の終点ノード

$rc_k$ :  $r_k$  の要求資源量,  $rc_k$  は正整数

$x_{i,j,k}$ :  $r_k$  における  $l_{i,j}$  への  $n_i$  から  $n_j$  方向への割当状況

$$x_{i,j,k} = \begin{cases} 0 & \text{割当なし} \\ 1 & \text{割当あり} \end{cases}$$

(1) 容量制約:

$$\sum_k rc_k (x_{i,j,k} + x_{j,i,k}) \leq lc_{i,j} \quad \text{for } \forall i, j$$

(2) 流量保存制約:

$$\sum_j x_{i,j,k} - \sum_j x_{j,i,k} = \begin{cases} 1 & : i = sn_k \\ -1 & : i = en_k \\ 0 & : i \neq sn_k, en_k \end{cases} \quad \text{for } \forall i, k$$

ここでの目的関数は  $z$  を最小化する  $x_{i,j,k}$  を求めるものであり、次式で表される。

$$\min z = \sum_{i,j,k} uc_{i,j} rc_k x_{i,j,k}$$

☆ コストを問題にしない場合は、使用コストを0とすればよい。

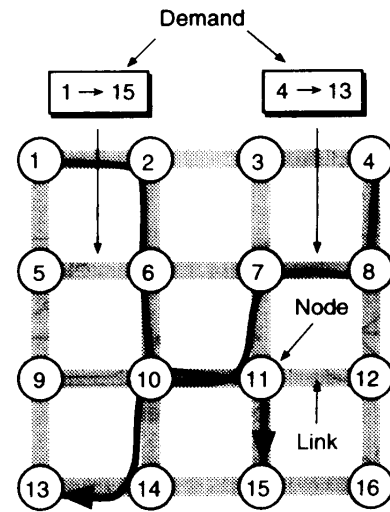


図1 資源割当問題

Fig. 1 Network resource allocation problem.

## 3. 協調問題解決手法

### 3.1 概要

提案手法は、ネットワーク資源割当問題を解く協調問題解決手法である。提案手法は、自律的な計算主体として複数のエージェントを用い、これらのエージェントが協調することにより問題解決を行う。エージェント間で資源の取り合いによる資源競合が生じた場合は、各エージェントは協調動作としてネゴシエーション（交渉）を行い、競合を解消する。

提案手法のシステム構成を図2に示す。本システムは、複数のエージェントと割当結果データベースから構成される。各エージェントはネットワークポロジヤやリンク容量に関するネットワーク情報を持っており、これを用いて自律的に資源割当を行う。割当結果データベースは、各エージェントが要求を1つ割り当てるごとにその結果を登録するもので、資源競合のチェックに用いられる。資源競合とは、同一のリンクに資源割当が集中し、その結果リンク容量を超える割当が行われた状態を言う。

提案手法は以下の3つの処理からなる。

- Stage1: エージェントへの要求の分配
  - Stage2: エージェントによる資源割当
  - Stage3: ネゴシエーションによる資源競合の解消
- ステージ1では、与えられた複数の要求を各エージェントに分配する。ステージ2では、各エージェントは、分配された要求をネットワーク情報を用いて個々に割り当てる。エージェントは、要求を1つ割り当てると、その結果を割当結果データベースに登録し、資源競合のチェックを行う。資源競合を発見した場合に

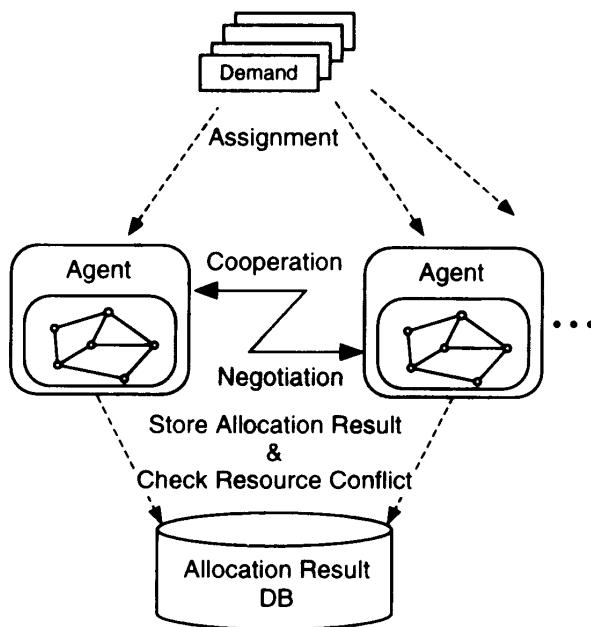


図2 提案手法のシステム構成

Fig. 2 Configuration of the proposed method.

は、ステージ3が実行される。ステージ3では、資源競合に関与しているエージェント間でネゴシエーションが行われ、資源競合が解消される。ステージ3は状況に応じて選択され、必ずしもすべてのエージェントがこのステージを実行するとは限らない。資源競合がなく、すべてのエージェントの資源割当が終了すると、割当終了となる。

提案手法は、このように各エージェントが資源競合を解消しながら非同期並行に資源割当を行っており、要求が動的に変化する場合にも対応可能な構成となっている。たとえば、要求の変更が生じた場合、その要求を受け持つエージェントのみが現行の割当経路を解放し、新たな要求として再度割当を行えばよい。

以下、各ステージの詳細を示す。

### 3.2 エージェントへの要求の分配 (Stage1)

与えられた要求を各エージェントに分配する。分配に際しては負荷分散を考慮し、各エージェントの分配要求数が均等になるように分配を行う。

要求が時系列的に発生するような動的な問題の場合には、要求が発生した時点で、未割当要求数の最も少ないエージェントから順に分配することで、均等割当を実現することができる。ただし、未割当要求数を把握するために、各エージェントは、要求を割り当てるごとに未割当要求数を要求分配部に報告する必要がある。

### 3.3 エージェントによる資源割当 (Stage2)

要求を分配されたエージェントは、それぞれ自律的

に資源割当を開始する。割当は要求の始終点間距離が短いものから順に行う。これは、最小経路リンクの経路数が少ない、すなわち低コストの経路候補が少ない、より条件の厳しいものから先に割当を行うためである。一般に始終点間の距離が短いほど、2点間を結ぶ最小経路リンクの経路数は少なくなる。経路コストは経路リンク数とそのリンク容量で決まるが、リンク容量のばらつきが小さい場合は、経路リンク数の少ない方が低コストになりやすい。

各エージェントは自身の持つネットワーク情報を用いてリンクの容量を超えないように、かつ使用コストが最小となる資源割当を行い、割当結果を割当結果データベースに登録する。

エージェントは、資源割当の際には他のエージェントの割当結果を参照しない<sup>☆</sup>ため、資源競合が起り得る。資源競合が起きた場合は、エージェント間でメッセージ交換を行い、後述するネゴシエーションによって競合の解消を行う (3.4 節参照)。

エージェント内部では以下の3つの処理を順に繰り返し行う (図3)。

#### Step1: 資源割当

エージェントは、始終点間の距離が最も短い未割当の要求を1つ選択し、Dijkstraのアルゴリズムを用いて使用コスト最小の経路を探索する。

#### Step2: 資源競合チェック

エージェントは、Step1で求めた経路を割当結果データベースに登録するとともに、資源競合のチェックを行う。資源競合を発見した場合は、競合を起こしている他のエージェントに対してネゴシエーション要求メッセージを送り、ネゴシエーションを行うことにより競合を解消する。

#### Step3: メッセージチェック

エージェントは、他エージェントからのネゴシエーション要求メッセージの有無を調べる。メッセージがある場合は、ネゴシエーションを開始する。メッセージがない場合は、未割当の要求を調べ、まだ要求がある場合はStep1へ、ない場合はメッセージ待ち状態に入る。

<sup>☆</sup> 資源競合が発生しないよう割当結果を逐次参照しながら今後の割当経路を決定する方法も考えられるが、この方法は競合防止の点では有効であるが、先に割り当てた経路が優先となり、解の質の点では必ずしも有効であるとは言えない。それでは割当結果を見た上で競合を起こすようにすればいいかと言えば、これは各エージェントが経路割当後に、結果を参照して競合の有無を調べる方法と同じとなる。したがって提案手法では、逐次参照のコストが必要ない割当後に一括参照を行う方法を採用した。

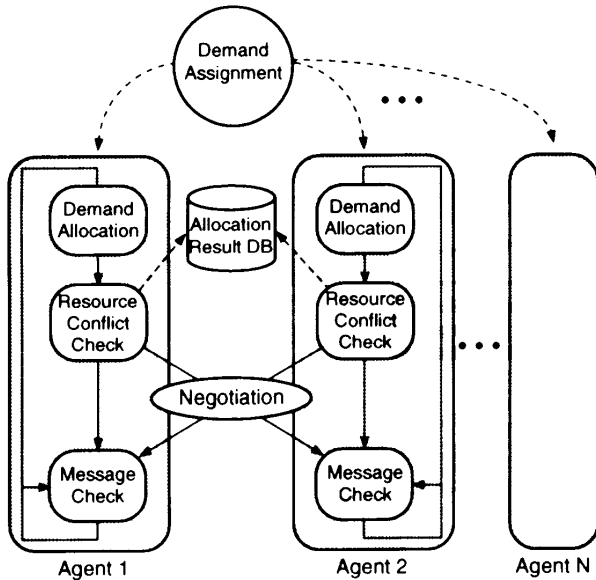


図3 エージェントの内部動作

Fig. 3 Internal operation of the agent.

### 3.4 ネゴシエーションによる資源競合の解消 (Stage3)

資源競合は、同一のリンクに資源割当が集中し、リンク容量を超える割当が行われたときに発生する。ここでは、容量を超える割当が行われたリンクを競合リンク、このリンクを含む経路を競合経路と呼ぶ。

資源競合が発生した場合、競合を起こしているエージェントの間で、協調動作としてネゴシエーションを行い、メッセージ交換により情報を収集し、競合解消のために迂回させる経路を決定する。ここでは、迂回させる経路を迂回要経路と呼ぶ。迂回要経路を受け持つエージェントは、割当結果データベースから迂回要経路の割当結果を削除し、競合リンクを除いたネットワークから、新しい経路を再探索する。

#### 3.4.1 ネゴシエーション手順

ネゴシエーションは、資源競合を最初に発見したエージェントが取りまとめ役となって行われる。第一発見者であるエージェントは資源競合を発見した後、他の競合相手にネゴシエーションを行うことを呼び掛け (Step1)、競合相手から競合経路の情報を収集し (Step2)、収集した情報に基づいて迂回要経路を決定し、結果を競合相手全員に通知する (Step3)。迂回要経路を受け持つエージェントは、迂回のために再割当を行う (Step4)。このような手順でネゴシエーションは進むため、迂回要経路決定に際してはエージェント間で1回のメッセージ交換 (情報収集と結果通知) が行われるのみである。

実際のネゴシエーション時のメッセージフローの例を図4に示す。この例では、エージェントAが最初

に資源競合を発見し、競合相手であるエージェントB、Cとネゴシエーションを行う。以下にネゴシエーション手順を示す。

**Step1:** 資源競合を発見したエージェントAは、競合を起こしているエージェントB、Cに対して、ネゴシエーション要求メッセージを送る。

**Step2:** ネゴシエーション要求メッセージを受け取ったエージェントB、Cはネゴシエーション状態に入り、競合経路に関する情報を局所解メッセージとして返送する。

**Step3:** すべての局所解メッセージを受け取ったのうち、エージェントAは、局所解メッセージを基に資源競合を解消するために迂回要経路を決定する。そして、その結果を判定メッセージとして再びエージェントB、Cに送る。

**Step4:** 判定メッセージを受け取ったエージェントB、Cは、その内容をチェックする。この例では、エージェントCの競合経路が、迂回要経路として選ばれており、エージェントCは該当する競合経路を割当結果データベースから削除し、迂回経路の探索に入る (再割当)。

#### 3.4.2 ネゴシエーション戦略

各エージェントは分配された要求を順に、その時点で使用できる資源を用いて使用コスト最小の経路に割り当てている。もし、すべてのエージェントの要求について資源競合なしに使用コスト最小の経路を割り当てることができれば、総使用コストの小さい準最適解<sup>\*</sup>を得ることが期待できる。しかし、実際には各エージェントは、他のエージェントの割当結果を参照せずに独自に資源割当を行っているため、資源競合が発生しうる。その場合、競合解消にともなう経路変更 (迂回) により一般に解の質は低下する。

このため、解の質の向上を目指した資源競合の解消戦略が必要である。我々は、ネゴシエーションにおける迂回要経路の決定戦略について検討した。検討した戦略は、迂回によって生じ得る新たな資源競合の低減をねらったものである。各経路は最初はエージェント内でコスト最小の経路が割り当てられており、通常は迂回するごとに、よりコストの大きい経路が割り当てられる<sup>\*\*</sup>。したがって資源競合を減らすことにより、

<sup>\*</sup> エージェントに分配された要求において、要求をどのような順序で割り当てても、各要求の最小コスト値が同じであれば、最適解となる。

<sup>\*\*</sup> 他の経路の迂回により資源が解放され、オリジナルの経路コストよりも迂回経路のコストの方が小さくなることもありうるが、こういったケースは少ないと考えられる。

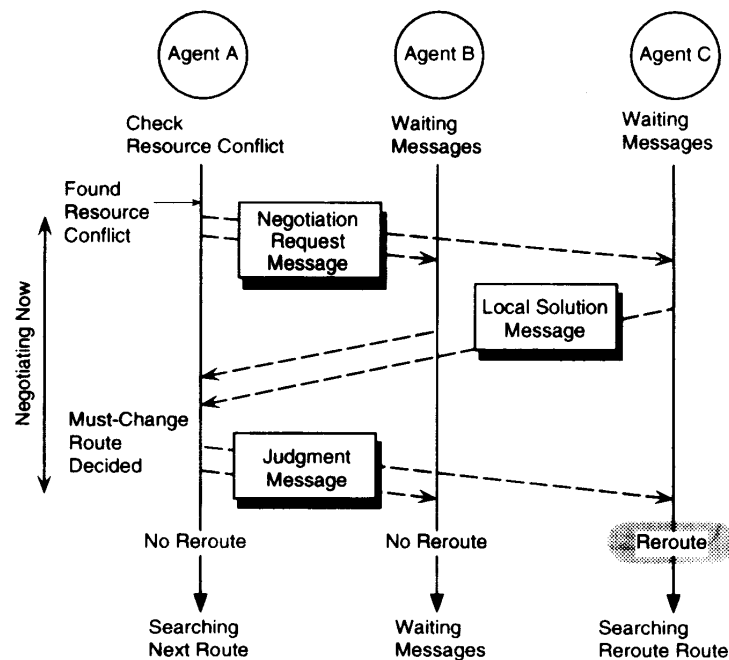


図4 ネゴシエーション時のメッセージフロー  
Fig. 4 Example of message flow during negotiation.

解の質向上の効果が期待できる。さらに、資源競合にともなうエージェント間のネゴシエーションも低減するため、処理時間の低減にも効果がある。我々は、ネゴシエーションにおける迂回要経路を決定するために以下の2つの戦略を用いた。

#### 戦略 A: 資源競合の低減

資源競合は容量の小さいリンクで発生しやすいため、容量の小さいリンクを多く使用している経路を優先的に迂回させる。

#### 戦略 B: 迂回経路の解の質向上

一般に始終点間の距離が長い要求ほど、元の経路と同じ経由リンク数の迂回経路が多い。また、リンク容量のばらつきが小さい場合には、経由リンク数の増加は、解の質低下につながる。したがって、経由リンク数が増加しにくい始終点間の距離が長い経路を優先的に迂回させる。

## 4. シミュレーションと評価

この章では、提案手法をネットワーク資源割当問題に適用し、シミュレーションにより集中手法と比較してその評価を行う。

### 4.1 シミュレーションの概要

#### 4.1.1 対象問題

ここではネットワーク資源割当問題の例題として公衆ネットワークの再編成を想定した回線の集約化を取りあげる。これは回線の経路設定の際にできるだけ容量の大きいリンクに経路を割り当て、回線を容量の大

きいリンクに集約化する問題である。2章で定式化した単位使用コスト  $uc_{i,j}$  を以下のように容量の大きいリンクほど小さくなるよう設定し、定式化を行った。

$$uc_{i,j} = \frac{1}{lc_{i,j}}, \text{ ただし } lc_{i,j} \neq 0$$

ネゴシエーション戦略の定式化には、経路コスト（経路内リンクの総使用コスト）を用いる。経路コストは、リンクの使用（戦略 A）と経路長（戦略 B）を反映している。戦略 A における迂回要経路は、容量の小さいリンクほど使用コストが大きいため、経路コストは大きくなる。一方戦略 B における迂回要経路も、始終点間の経由リンク数が多くなるために経路コストは大きくなる。したがって、競合が解消するまで、経路コストの大きい経路順に迂回要経路とすることにより両方の戦略を満たしたネゴシエーション戦略が実現できる。

実際のネゴシエーションでは、各エージェントは各自の競合経路の経路コストを計算し、その結果を局所解メッセージとして資源競合を発見したエージェントに送る。資源競合を発見したエージェントは、各エージェントから送られてきた経路コストを比較し、競合の解消に必要な迂回要経路を、経路コストの大きい経路順に決定する。この結果を、判定メッセージとして各エージェントに送る。

#### 4.1.2 シミュレーション条件

本シミュレーションでは、ネットワークとして5種類のリンク容量（10, 15, 20, 25, 30）が一様分布す

るノード数  $N \times N$  の格子状ネットワークを用いた。また、要求は要求資源量をすべて5とし、始終点ノードをランダムに選んで生成した。ここでネットワークに対する要求の量を示す指標として、要求の割合  $Rate_D$  を以下のように定義した。ただし、 $l_{min_k}$  は要求  $r_k$  における始終点間の最小経由リンク数とする。

$$Rate_D = \frac{\sum_k l_{min_k} \times r_{c_k}}{\sum_{i,j} l_{c_i,j}}$$

また、エージェント間のメッセージ通信に転送遅延が生じるものとし、転送遅延時間  $T_D$  を設定した。

#### 4.1.3 解の存在範囲

本シミュレーションでは要求をランダムに生成するため、必ずしも解が存在する（すべての要求について割当が可能である）とは限らない。そこで、要求の割合に対する解の存在確率を調べた。解の算出は、2章で定式化したネットワーク資源割当問題に分枝限定法を適用し、 $4 \times 4$  の格子状ネットワークに対して行った（図5）。要求の割合が0.5を超えるあたりから解の存在確率が急激に低下し始め、0.8以上ではほとんど解が存在しない結果となった。そこで、要求の割合  $Rate_D$  を0.5としてシミュレーションを行うことにした。

#### 4.1.4 シミュレーション環境

このようなシミュレーション条件のもとで、C++言語を用いて提案手法、および比較用として集中手法（近似解法）のプログラムを作成し、SUN SPARCstation 2上でシミュレーションを行った。

集中手法には、始終点間の距離が短い資源要求から順に、これまでに割り当てた資源は利用できないものとして、使用コスト最小経路を探索していくという欲張り法を用いた。この手法は、再割当処理を行わない単純な手法のため、集中方式の近似解法では最も高速

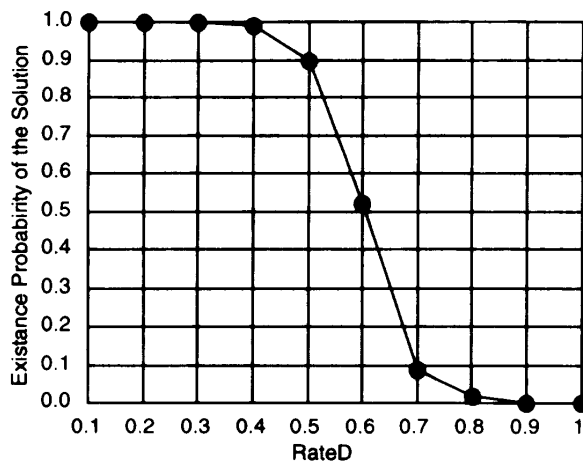


図5 解の存在確率

Fig. 5 Existence probability of the solution.

な部類に入ると思われる。

シミュレーションでは、DBを共有メモリ上の変数として表現しており、変数へのアクセスにセマフォを用いて排他制御（同時アクセス制御）を行った。また、各エージェントのDBアクセスのコストは探索時間に比べて無視できるものと仮定した。

## 4.2 シミュレーション結果

### 4.2.1 ネットワーク規模の処理時間への影響

集中手法および提案手法に対して、 $Rate_D=0.5$ ,  $T_D=0$  msec, 100 msec, エージェント数  $N$  として、ノード数  $N \times N$  を変化させた場合の処理時間を図6に示す。

この結果は、規模の大きいネットワークにおける提案手法の有効性を示している。集中手法の処理時間は、ノード数の自乗に比例して増加していくが、提案手法の処理時間は、メッセージ通信に遅延時間がある場合（ $T_D=100$  msec）であっても、緩やかに増加しており、ノード数が増えるほど集中手法に比べて処理時間を低減することが可能である。

しかし、ノード数が64以下の小規模なネットワークの場合は、遅延時間（ $T_D=100$  msec）のある提案手法の処理時間は、200~300 msecほど集中手法に比べて増加する結果となった。これはノード数が小さい領域では経路探索にかかる時間よりも、メッセージ通信の遅延時間の方が大きいと思われる。

以上の結果から、ノード数が多くなるほど、すなわちネットワーク規模が大きくなるほど提案手法が有効となる。

### 4.2.2 転送遅延の処理時間への影響

ノード数  $10 \times 10$ ,  $Rate_D=0.5$ , エージェント数 10

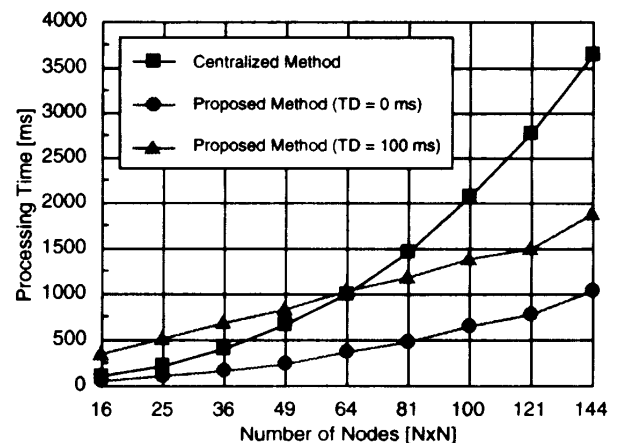


図6 ノード数と処理時間

Fig. 6 Processing time depend on network scale ( $N$  agents,  $Rate_D = 0.5$ ).

として、転送遅延時間 $\star$ を変化させた場合の処理時間を図7に示す。この図では、処理時間は3つの部分－探索時間、ネゴシエーション時間、アイドル時間－に分かれている。探索時間は、経路探索と割当結果データベースへの登録に要する時間、ネゴシエーション時間は、ネゴシエーションに要する時間、アイドル時間は、メッセージ待ちなど、割当もネゴシエーションもしていない時間である。これらの時間は、1エージェントの平均時間である。

この結果は、転送遅延時間の増加は、ネゴシエーション時間の増加だけでなく、アイドル時間も増加することを示している。これは、ネゴシエーション中のエージェントにネゴシエーション要求を出した場合、ネゴシエーションが終了するまで待つ必要があるが、ネゴシエーション時間の増加にともなって待ち時間も増加するからである。このように、転送遅延時間の増加はネゴシエーション時間だけでなく、待ち時間の増加にも影響する。

ここで待ち時間に注目してみると、待ち時間は転送遅延時間が0 msecの場合においても処理時間の半分以上を占めている。待ち時間はエージェントが何もしていない状態の時間であり、待ち時間が大きいとエージェントの稼働率が低下する。したがって、この待ち時間を減少させることが今後の課題である。

#### 4.2.3 エージェント数の処理時間への影響

ノード数 $10 \times 10$ 、 $Rate_D = 0.5$ 、 $T_D = 100$  msecとして、エージェント数を変化させた場合の処理時間を図8に示す。この図も、処理時間は、探索時間、ネゴシエーション時間、アイドル時間の3つに分けて示している。

この結果から、エージェントを増やすと、1エージェントあたりの探索時間およびネゴシエーション時間は減少するが、待ち時間は増加することが分かる。このため、エージェントを増やすほど処理時間が低減するわけではなく、問題に応じて処理時間が最小となる最適なエージェント数が存在すると思われる。

図8の例では、エージェント数が10程度で処理時

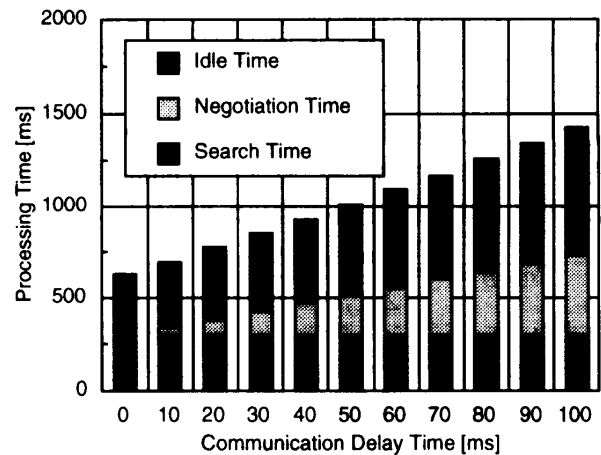


図7 遅延時間と処理時間

Fig. 7 Processing time depend on communication delay time ( $10 \times 10$  nodes, 10 agents,  $Rate_D = 0.5$ ).

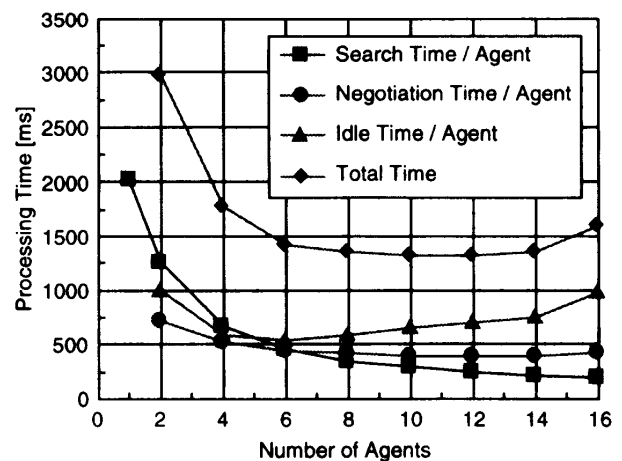


図8 エージェント数と処理時間

Fig. 8 Processing time depend on the number of agents ( $10 \times 10$  nodes,  $Rate_D = 0.5$ ,  $T_D = 100$  ms).

間が最小となっている。さらに資源要求の割合を0.5に固定して、ノード数 $N \times N$ を変化させて同様のシミュレーションを行った結果、概ねエージェント数が $N$ のときに処理時間が最小となった。また、このときの1エージェントあたりの要求数は4~6であった。一方、ノード数を $10 \times 10$ に固定して、資源要求の割合を変化させた場合は、1エージェントあたりの要求数が5~6のときに処理時間が最小となった。

以上の結果から、このシミュレーションの問題における最適なエージェント数は、1エージェントあたりの要求数が4~6になるようなエージェント数である。

#### 4.2.4 解の質

ノード $4 \times 4$ 、エージェント数4として、要求の割合を変化させた場合の解の質を図9に示す。解の質は、提案手法および集中手法で求めた解のコスト値を、分枝限定法による最適解のコスト値（最小コスト値）で

$\star$  実際の転送遅延時間はエージェント間の通信網の転送速度や使用状況によって異なる。ここではイーサネットに接続されたWSを想定し、TCP/IPによるデータ転送の転送遅延時間を測定した。その結果、100 byteのデータ転送における転送遅延時間は10 msec程度であった。イーサネットの最大転送速度は約1 Mbyte/sec<sup>11)</sup>であり理論上の転送遅延時間は0.1 msecとなるが、実際にはネットワークの混み具合や、通信路の確立等に時間がかかるため、100倍程度の値になる。提案手法のメッセージは最大でも100 byte程度であり、メッセージ通信による転送遅延時間は、10 msec~数10 msecと思われる。

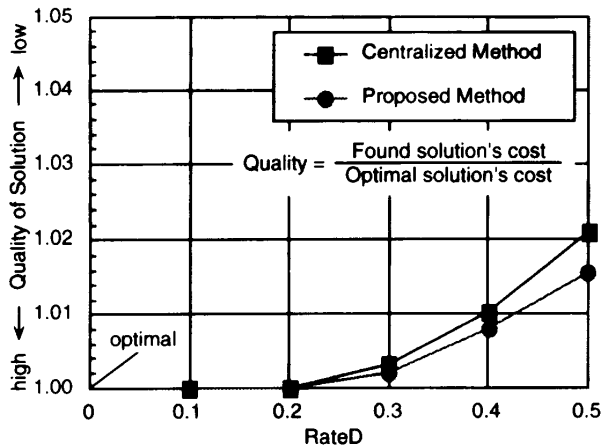


図9 要求の割合と解の質

Fig. 9 Quality of solution (4x4 nodes, 4 agents).

割ったもので表した。

提案手法の解の質は、要求の割合が増加すると低下するが、最適解に比べて0.98~0.99の近似解が得られた。また、提案手法の解の質は、集中手法と比べて良い結果となった。この理由として、提案手法では迂回により再割当が行われることが考えられる。集中手法では、これまでに使用した資源は使えないものとして始終点間の短い要求から順に割当を行っている。したがって、割当の順序が後になるほど使える資源の制約は厳しく、割当順序が初めの方の場合よりも解の質は低下しやすい。一方、提案手法では、各エージェント内でコスト最小経路を割り当てており、競争が生じた時点で、いずれかを迂回させるということで競争の解消を行っている。すなわち、順に割当を行い、その時点での最小コスト解を割り当てていく方法よりも、最小コスト解で割当を行ってみて、競争が生じた時点でいずれかを再割当させる方法の方が解の質の面で効果的であったと言える。

提案手法の処理時間を分枝限定法と比較する。ネットワークサイズ4x4、要求の割合0.5の問題40問における、分枝限定法および提案手法の平均処理時間は、それぞれ1540秒、0.3秒であった。この結果は、提案手法が分枝限定法に比べ約5000倍程度速いことを示している。また、ネットワークサイズがこれ以上になると分枝限定法の処理時間は数時間必要となり、測定は断念したが、この差はさらに大きくなると容易に予想される。

以上の結果から、分枝限定法に比べて十分速く、かつ最適解に近い近似解が得られる提案手法は有効であると言える。

## 5. おわりに

本稿では、ネットワーク資源を経路設定などの要求に割り当てるネットワーク資源割当問題を定義し、その解決手法として複数のエージェントが互いに協調して問題解決を行う協調問題解決手法を提案した。

提案手法は、大規模ネットワークや要求などが動的に変化する場合、さらには要求ごとに制約があるような場合にも有効であるように、要求を各エージェントに分配し、分配された要求を受け持つ複数のエージェントが協調することにより問題解決を行う手法である。本稿では個々のエージェントによる資源割当メカニズムを示し、資源競争発生時にエージェント間で行う協調動作として、処理時間の低減と解の質向上を目的としたネゴシエーション戦略を検討した。

提案手法を公衆ネットワークの再編成を想定した回線の集約化問題に適用し、シミュレーションにより処理時間および解の質について集中手法と比較/評価を行った。シミュレーションでは転送遅延時間やエージェント数の変化に対する処理時間への影響についても議論した。その結果、ネットワークの規模が大きい領域においては、提案手法は、質の高い近似解を集中手法の半分以下の処理時間で求めることができ、大規模ネットワークにおける提案手法の有効性を確認した。

今後、待ち時間低減のための要求分配法およびネゴシエーション戦略の検討、要求などが動的に変化する場合の評価を行う予定である。

**謝辞** 本研究の機会を与えて下さったNTTコミュニケーション科学研究所前河岡司所長、研究頭初から多くのコメントをいただいたNTTコミュニケーション科学研究所の桑原和宏主任研究員、NTT通信網総合研究所の伊藤大雄主任研究員に感謝いたします。

## 参考文献

- 1) 辻井重男, 黒沢 馨: 通信網における多種フロー問題の発見的アルゴリズム, 信学論 (A), pp.825-832 (1979).
- 2) 翁長健治, 有本厚志, 岸本一男: ネットワークにおける多種結線実現問題とその近似算法, 信学論 (A), pp.431-438 (1982).
- 3) Bond, A.H. and Gasser, L.: *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers (1988).
- 4) 石田 亨, 桑原和宏: 協調問題解決, 人工知能学会誌, Vol.7, No.6, pp.13-21 (1992).
- 5) Conry, S.E., Kuwabara, K., Lesser, V.R. and Meyer, R.A.: Multistage Negotiation for Distributed Constraint Satisfaction, *IEEE Trans.*



*Systems, Man, and Cybernetics*, Vol.21, No.6, pp.1462-1477 (1991).

- 6) 湊 賢治, 奥村康行, 岸本了造: 分散協調マルチエージェントモデルによる通信網の複数回線設定制御, 信学論 (B), pp.909-918 (1991).
- 7) 村山隆彦, 和氣弘明, 土田尚純, 服部文夫: 資源割当問題向き協調問題解決手法の検討, 情処 45 全大 (1992).
- 8) 和氣弘明, 村山隆彦, 土田尚純, 服部文夫: 資源割当問題向き協調問題解決手法, 情処 46 全大 (1993).
- 9) 和氣弘明, 村山隆彦, 土田尚純, 服部文夫: 資源割当問題向き協調問題解決手法, 信学技報, AI93-71, pp.91-98 (1993).
- 10) Waki, H., Murayama, T. and Hattori, F.: Cooperative Problem Solving Suited for Network Resource Allocation, *Proc. of EXPERSYS-94*, pp.131-136 (1994).
- 11) Stevens, W.R. (篠田陽一訳): UNIX ネットワークプログラミング, トッパン (1992).
- 12) 桑原和宏, 石田 亨: 分散資源割当における共生的アプローチ: 通信網の資源割当を目指して, 信学技報, AI92-67, pp.41-48 (1992).
- 13) H.M. ワグナー (森村英典, 伊理正夫監訳): オペレーションズ・リサーチ入門, 培風館 (1986).
- 14) 青木武司, 泉 寛幸: 通信ネットワークのコスト最適設計の高速化, 人工知能学会誌 (1991).

(平成 7 年 7 月 19 日受付)

(平成 7 年 12 月 8 日採録)



和氣 弘明 (正会員)

昭和 41 年生. 平成 3 年豊橋技術科学大学大学院工学研究科情報工学専攻修士課程修了. 同年日本電信電話(株)入社. 知識処理システム, 分散人工知能システムの研究に従事. 現在, エージェント通信に興味を持つ.



村山 隆彦

昭和 59 年東北大学工学部通信工学科卒. 昭和 61 年同大学大学院工学研究科情報工学専攻修士課程修了. 同年日本電信電話(株)入社. 以来, エキスパートシステム, 知識処理システム, エージェント指向システムの研究開発に従事. 電子情報通信学会, 人工知能学会, 日本ソフトウェア科学会各会員.



服部 文夫 (正会員)

1950 年生. 1973 年早稲田大学理工学部電子通信学科卒業. 1975 年同大学大学院理工学研究科修士課程修了. 同年日本電信電話公社入社, 横須賀電気通信研究所勤務. データベースシステム, 知識ベースシステムなどの研究開発に従事. 現在, NTT 情報通信研究所主幹研究員. エージェント通信およびその応用に興味を持つ. 電子情報通信学会, 人工知能学会, AAAI 各会員.