

観測差とそれに基づく協調方式について

菅原 俊治

NTT ソフトウェア研究所
〒180 武蔵野市緑町 3-9-11
sugawara@ntt-20.ntt.jp

概要

単独で解決できる問題と協調して解くべき問題が混在した分散問題を考える。問題解決では、問題解決の知識と問題の絞り込みを可能とするフォーカスが必要である。問題の混在は、その場合に応じて、知識とフォーカスが同一エージェントまたは異なるエージェントに存在することにより起こる。ここでは具体例としてインターネットワークの診断問題を取り上げる。ネットワークでは、障害原因があるエージェントの管理範囲にあるにもかかわらず、障害の認知やデータの収集ができないという問題がある。一方、他のエージェントは障害発生の可能性を推測できるが診断はできない。これは知識とフォーカスが分離した典型例である。このような問題では、障害発生を知識を持つエージェントになるべく早く認知させ(フォーカスを与え)、全体の主導権を握ることが迅速な推論を実現する。本論文では、単純な実験を通して、単独/協調の混在の問題点を洗いだし、統一して扱える協調問題解決のコントロールを追究することにある。

Cooperation Based on Difference of Observations

Toshiharu Sugawara

NTT Software Laboratories

ABSTRACT

We consider the case that two types of distributed problems, which should be resolved autonomously or cooperatively, alternately appear. For problem-solving, knowledge for diagnosing the problem and focus which can restrict the range of the problem are necessary. Such a mixture is caused by which knowledge and focus is in an agent or in other agents. Diagnosis of internetwork problems are examples of this kind of problems. In internetworking, the focus of the problem sometimes exists in the agent which is physically or logically far from the network segments where the cause is. Thus, if the agent having knowledge finds that he cannot resolve it autonomously, he must contact the agent having focus as soon as possible. Through actual experiments, uniform control strategies which can treat these two types are discussed.

1 はじめに

分散システムによる問題解決とは、各エージェントがそれぞれの視点と知識から全体として整合性のとれた解を見つけることである。ここで問題が完全に分割可能な場合と、分割不能であり全体として consistent な解を求める場合がある。前者では、独立したプロセスの集合と考えられ、並列化、高速かなどが主な論点となる。後者は、協調推論を必要とし、主に分散人工知能(DAI)と呼ばれる分野で、ネゴシエーション、プランニング等を主な話題として活発な研究が行われている。

これまで DAI 研究用の TESTBED やアプリケーションとしていくつかのシステムが提案されてきた [2, 3, 1, 4]。これらは、物理的 / 論理的に分散されたエージェントが、それぞれの不完全な情報とエラーを含むかもしれないデータから解の候補を作りだし、全体として整合性のとれたグローバルな解を求めることにある [5]。仮に 1 エージェントで有力な結果が得られたとしても全体として受け入れられるものでなければ、それは却下される。しかし、これらは予め協調推論が必要であることを仮定している。

本論文では分割可能な問題と不能な問題が混在した問題を対象とする。このような場面では、初めに問題の「型」を認知し、単独で解く問題か協調して解く問題かを判定しなくてはならない。ここでは、さらに、問題の型を認知すること自体も困難な場合を考える。したがって、問題の型の判定も不確実であり、一時的に単独で解く問題と判定しても、推論が進むにつれてこれが覆ることがある。また、問題の型を定める情報も一エージェントだけでは不完全なこともある。単独 / 協調の両者の型にも共通して使えるコントロールが必要となる。

ここでは、インターネットワーク監視 / 診断システム LODES (Large internetwork Observation and Diagnostic Expert System)[6, 7]を通じて、このような問題の効率的な推論 / コントロール方式を求めることを目的とする。本論文では、はじめに問題点とそれに要求される条件についての、具体的な実験を通してさらに細かく解析する。つぎに、プラン通知、診断状態数等を導入し、それらの効果を実験を通して評価する。

2 問題の型と認知

分散協調問題解決は複数のエージェントが、それぞれの知識 / 物理的または論理的配置の違いなど活かしながら、互いの情報を交換し問題を解決することである。ここでは、予め協調が必要であることを前提としている。しかし実際の場面においては、あるエージェントが単独で解くべき問題とエージェント達が協調して解くべき問題が混在していることがある。

このような場面では初めに問題を認知し、その問題がいずれの型に属するかを定め、それに応じて単独 / 協調して問題を解くべきである。たとえば、エージェント A と B がこのような問題を解くとしよう。可能性として、(1) A が単独で解ける場合、(2) B が単独で解ける場合、(3) A と B が協調して解く場合がある。直面している問題どの部類に属するかを区別し、問題を解くべきである。ここで扱う問題は、さらに以下の二つの特徴を持ったものである。第一に、直面した問題を正確に区別することが困難である。仮に (1) のタイプと判断しても、将来、(2) または (3) の型と訂正されるかもしれない。(1) のタイプと判定し、その間 B の推論を停止させてしまうのは、グローバルな立場から効率的とは言えない。B はできる限りのことを進めておくべきである。

第二の特徴は、問題の型の判断自体にも、A と B との協調が必要である。例えば、(1) や (2) の問題であれば、A または B が独自にその問題の型を推定できる。しかし、(3) の型に属していると、A も B も問題の存在すら認知することができない。この場合には A と B の協調が本質的になる。この他に弱協調型の問題として、たとえば A 単独でも診断ができるが、B からの情報があると推論を効率化できることもある。

ここで求められるのは、問題の型に依存しない統一的なコントロールを導入することである。仮にエージェント A が (1) の型と判断すれば A は自己の推論を優先し、エージェント B からの通信をある程度無視すべきである。ここで B の推論を停止させることも考えられるが、問題の型の推論が覆ったときを考えて、A の推論を妨げない程度に進めておくべきである。また (2) と判断すれば、協調推論がスタートするが、B 側では A からの通信をより優先術機である。A の通信により (2) と B も判断すれば、B も協調主体とし、同意

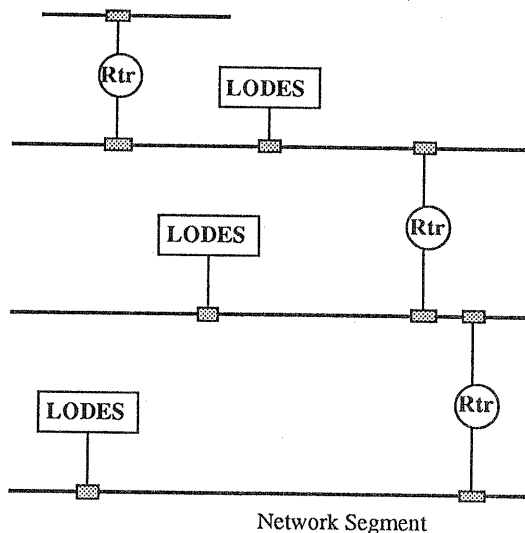


図 1: LODES の配置図

Fig.1. Global Network Configuration

できなければ A 独自の診断を進める。さらに (3) なら A は B にその根拠を示し、B に問題の認知をさせなくてはならない。

プラン選択の確信度がここでは重要なキーと考えられる。問題の型を認識できないことは、ある程度の問題の絞り込みができたことを意味し、したがって、問題を解ける可能性が高いと考えられる。一般にマルチエージェントシステムでは、コントロールの際にプラン情報を利用するのが効果的である。相手のプランを知り、全体として有益となるアクションを起こす。しかし、本問題で重要なのは、そのプランを選択した根拠である。たまたま記述された順番にて選ばれたのか、有力な根拠があって選ばれたのかによって、相手側がそれに従うべきか否かが定まる。

3 問題

本節では具体的な問題を説明と実験に必要な LODES の説明を行う。

3.1 LODES の概要

LODES はインターネットワークで発生する障害を発見し、原因究明を行う知識ベースシステムである。図 1 に示すように、各ネットワークセ

グメントが LODES エージェントを持ち、それらが流れるパケットを常時監視し問題を自動発見する。障害が発生すると問題に応じて単独もしくはは協調して診断を行う。LODES は、ネットワークの管理データ¹を基に、以下の手法により診断を行う。

- パケットを収集し解析を行う。
- 特定のパケットをホスト / ルータ等に送り、その応答を観測する。
- ホスト / ルータから管理情報 / トラフィック等の情報を収集する

また協調した診断には、特定の箇所ではしか収集できない情報を要求 / 送付する他、

- パケットの多点同時収集、
- 多点で収集されたパケットの付合せ、
- 観測点の違いによる観測結果の差違の発見、

といった手法が使われる。これらの協調を実現するために診断中に LODES 間で以下のメッセージの転送が行われる。

- データの要求 — 自己の診断に必要なデータを他のエージェントに要求する。
- データの送付 — 他のエージェントにとって必要と推測されるデータを転送する。
- JOB の依頼
 - 自己の診断に必要なデータが相手側で求まっていないとき、JOB の依頼をだしてこれを求める。
 - 相手側に必要と思われる作業を依頼する。
 - 協同作業、たとえば同時多点パケット収集などの依頼などを依頼する。
- 時間合わせ (同期) — 同期が必要な協同作業のための同期を合わせる。

¹たとえば、IP アドレスと MAC(Media Access Control) 番号の対応表、各ホストが提供するサービス表、最近の各ホストのアクティビティ、ネットワークの平均的なトラフィック、MAC 番号と計算機メーカー対応表、SRI-NIC(Network Information Center) より正式に取得しているインターネットアドレスの一覧表など。これらのほとんどは自動収集される。

3.2 LODES における推論

LODES は Local Planner (LP), Communication Scheduler (CS), Controller (CTL), Rule Interpreter (RI) からなる。LP は各エージェントの知識と観測データをもとにプランを選択する。プランはルールが TREE 状に結ばれたもので、ルールには対応するオペレーションが記述されている。ルールインタープリターがルールを実行すると、始めにオペレーションが実行され、その結果をもとに、次のルールまたはプランを CTL に提案する²。CS は他のエージェントからの JOB の依頼とデータ転送を受け、これらの実行/参照 (これにもオペレーションとその結果に基づいて次のルールやプランを選択するルールが対応している) を CTL に提案する。CTL は他のコンポーネントから提案されたルール (またはルールに対応したオペレーション) を何らかの方法で選択し、それを実行する。

LP によって選択されたプランと、CS によって提案されたオペレーションの結果に基づいて選択されたプランとがある。それぞれのプランによって CTL に提案されたルールの集合を OS_L , OS_R と表す。

各ネットワークはそれに属するホストとルータも含め、約 100 項目の属性と属性値として表現される。この属性値の集合をネットワークのモデルと呼ぶ。各属性値は LODES に定義されているオペレーションを実行して求まる。

診断に参加している LODES エージェントは、各が管理するネットワークセグメントと、診断に必要と思われる他のネットワークセグメントのモデルを構築し、正常な場合のモデルとの食い違いから原因を判断する。しかし他のネットワークセグメントのモデルを完全に作ることはできない、つまり一部の属性値のみが明らかとなる。これは、他のネットワークセグメントの観測は、たとえばルータなどを介した結果であり、そのためパケットの一部が変化したり、消失することもあるからである。またセグメント内に閉じたパケットのやりとりも当然観測できない。この結果、真に原因究明が可能となるのはその原因の位置する LODES エージェントである。

²したがって、通常のプロダクションルールとは異なる。プランごとにもジュール化された、より手続き的表現に近い形となっている。

3.3 問題

原因究明が可能なエージェントは、その原因の位置するもののみであるにもかかわらず、そのエージェントから問題を検出できないことがある。たとえばあるホストでのルーティングの設定ミスが単純な例である。同一セグメント内のパケットに対しては常に正しい応答を示す。したがって、その LODES エージェントからはすべてが正常に動作しているように見える。一方、この障害例では、他のセグメントの LODES エージェント、特に設定ミスの対象となっているネットワークからは、パケットに対してもこのホストからは応答がないように観測されるが、これから直ちに結論は導けない。たとえば、パソコンやなどでは正常でも応答を示さないものがあり、他にパケットの種類やサービスの内容によってはホストもしくは途中のルータの設定により正常状態で無応答となることもある。また、相手側に障害があるとしても、ルーティングの異常の他に、ハード異常 (断線、インターフェースボードの故障、電源 OFF など)、UNIX マシンであれば single user mode で立ち上がっている場合、途中のルータやネットワークに異常がある場合などが考えられる。

問題の状況に応じて最適な協調方法が変わる。たとえば、ルーティングの設定ミスが障害の原因であるなら、相手からの観測結果をなるべく早期にもらい、それに対応する自己の値を比較 (観測差) しなければ、問題の認識すらできない。ハード異常の場合には、単独で問題の発見ができ、不要な通信は混乱を招く。また Single User Mode が障害の原因であれば単独でも発見はできるが、相手からの観測結果があると早くフォーカスが定まり、診断の時間が短縮する。途中のルータの設定による特定サービスの遮断については、相手からの観測結果の他、途中のネットワークに位置する LODES エージェントの協調が必要となる。したがって、さまざまな問題に対し、効果的な協調を実現するためのコントロールを実現しなくてはならない。

3.4 客観的観点から

この問題を客観的に解析する。問題の解決あるいは問題そのものを認識するためには、知識とフォーカスが必要である。知識とは、たとえ

ば診断問題では障害とその理由を記述した空間(探索空間)であり、一方フォーカスは、その探索空間を制限し推論を事実上可能とする情報である。LODESでは、仮に問題があるセグメントに属していたとすると、その問題を最終的に診断できるのはそこに位置するエイジェントのみである。つまりそのエイジェントが知識を持つことになる。これは問題に依存しない。フォーカスについては、たとえばハード異常のように単独で問題が解ける場合は、同一のエイジェントに知識とフォーカスが存在し、ルーティングの問題では他のエイジェントがフォーカスを持っている。Single User Modeの問題は、同一エイジェントが知識とフォーカスを持つが、他のエイジェントのフォーカスの方がより精密であるか推論過程の早期に得られる場合である。

これまで述べてきた問題の認知とは、フォーカスを得ることと同値である。ただし、一時的に得られたフォーカスが常に正しいとも限らない。フォーカスの変化が問題の「型」の変更に相当し、それに伴い診断の形態も単独/協調から協調/単独へと変わる。

4 実験 I

4.1 実験内容

はじめに二つの障害診断を例に実験を行う。ともに、二つのLODESエイジェントAとBが協調し、障害の原因はB側のネットワーク(NB)に位置するものとする。両問題とも障害はA側のユーザーから、B側のあるホストと通信ができない、という苦情があったとする。エイジェントAからは如何なる観測もこれらの問題において差はでない。第一の問題(PA)は、ハード異常の例で、原因の位置するLODESエイジェントBが単独で診断できる。第二の問題(PB)は、ルーティングの異常で、エイジェントBからの観測だけではすべてが正常に動作しているように見える場合である。このときにはエイジェントAからのフォーカスをえるためのデータの転送と問題を認識させるためのJOBの依頼、さらに障害観測のための同期したパケット収集が必要となる。

診断が進行するにつれて各エイジェントはデータの要求/送付とJOBの依頼を行う。これらを受けた側では、LOCAL PLANによって選ばれたオペレーション(各ルールにはオペレー

ションが対応している)の集合(OS_L)と相手側から送られてきた情報の参照とそれに伴うオペレーション集合(OS_R)の中から、コントロール部が適切なものを選択し実行しなくてはならない。第一の実験では、問題点を明らかにするために、機械的な次の三つのコントロール方式を考える。

- C1: OS_L を優先し、 OS_L から特に選ぶべきオペレーションがないとき OS_R から選択する。
- C2: OS_L と OS_L から交互にオペレーションを選択する。
- C3: C2と同じだが、要求されない限り相手の診断に必要なと思われるデータの送付やJOBの依頼を行わない。

これらは単純なコントロールであるが、本質的に求められている問題を知るには有効である。

4.2 実験結果

本実験の結果を表1,2にまとめる。ここでCTLはコントロールの種類、 OS_L , OS_R は各エイジェントがLocal Planerまたは相手側の要請によって選んだオペレーションの数、DR, DTは自己の推論のために必要なデータの要求または相手の推論に必要なと思われるデータの転送の数、 JR_1 , JR_2 , JR_3 は相手エイジェントに対するJOBの要求数で、順に自己の推論のために必要なJOB(JR_1)、相手の推論に必要なと思われるJOB(JR_2)、同期をとる協同作業(JR_3)を表す。C0の欄は適切な作業の実行数を示す。

問題PAではエイジェントBが問題の知らせを受ければ単独でも診断できる障害であり、A側からのデータ転送(DT)、協同作業の要求(JR_3)、B側の立場では相手側からの情報により選ばれたオペレーション(OS_R)は不要である。問題PBはエイジェントAからの観測結果がないとエイジェントBは診断できない問題である。A側からのデータ転送のうちの1つが本質的であり、それをBが認識した後、協同作業と試験を繰り返して障害原因が判明する。なお、B側からのデータ要求(DR)の内4つは、障害の症状を得るために必須であり、各問題においてこの部分は不変である。

問題PAではコントロールC1とC3が最適の

表 1: 問題 PA の場合

Ctl	A 側から							B 側から						
	OS _L	OS _R	DR	DT	JR ₁	JR ₂	JR ₃	OS _L	OS _R	DR	DT	JR ₁	JR ₂	JR ₃
C0	6	0	0	3	0	0	1	7	0	4	1	0	0	0
C1	10	1	0	3	0	0	1	7	0	4	1	0	0	0
C2	10	1	0	3	0	0	1	7	3	4	2	0	0	0
C3	10	1	2	0	0	0	1	7	0	4	0	0	0	0
C4	10	0	0	2(3)	0	0	1	7	0(3)	4	0	0	0	0
C5	10	0	0	3	0	0	1	7	0	4	0	0	0	0
C7	10	0	0	3	0	0	1	7	0	4	0	0	0	0

表 2: 問題 PB の場合

Ctl	A 側から							B 側から						
	OS _L	OS _R	DR	DT	JR ₁	JR ₂	JR ₃	OS _L	OS _R	DR	DT	JR ₁	JR ₂	JR ₃
C0	6	2	0	2	0	0	1	5	6	4	1	1	1	2
C1	10	3	0	4	0	0	1	7	6	4	3	1	1	2
C2	10	3	0	4	0	0	1	7	6	4	3	1	1	2
C3	10	3	2	0	0	0	1	7	6	6	0	1	1	2
C4	10	3	0	4(3)	0	0	1	7	6	4	1	1	1	2
C5	7	2	0	4	0	0	1	7	6	4	1	1	1	2
C6	7	2	0	4	0	0	1	6	6	4	1	1	1	2
C7	7	2	0	4	0	0	1	5	6	4	1	1	1	2

実行履歴を残す。実際、以降の実験を含めても問題 PA に対しては C3 が最も推論時間が短い。これは単独で診断できる問題に対し、相手の情報を後回しにするという盲目的なコントロールがたまたま適合したにすぎない。C2 では単純に交互にオペレーションが選択されるため、相手から届いた情報による無駄なオペレーションが実行される。問題 PB には、相手からの情報に本質的なものを含む。したがって、C1、C3 より C2 の方が問題を早く認識できる。しかし実際にはその効果は現れていない。これは交互にオペレーションを選択するため、B 側で LP により選ばれた本質的でないオペレーションも実行されるからである。なお C3 は C1 より推論時間がかかる。相手から情報が伝わらないために、自分にとって必要なデータを自ら解析し、要求するからである。

C1 - C3 のコントロールは盲目的である。機械的にオペレーションを選択しているにすぎない。ここで観測される協調の問題は以下の通りである。

- (1) 共に A 側で実行されるオペレーションのうち 4 つは不要である。問題 PA では B が単独で問題究明ができるため実害はない。しかし問題 PB では B が問題を認識した後、JOB の要請 (JR₁, JR₂, JR₃) を行い、これが A 側ですみやかに実行されなくては行けないが、不要なオペレーションの実行がこれを妨げる。
- (2) C2-PA では B 側において相手からの不要なデータにより無駄なオペレーションが実行され診断が遅れる。
- (3) C2-PB では問題の認識ができたにもかかわらず、不要なオペレーションの選択が発生する。
- (4) 問題 PB において、B の問題の認識は A による観測結果とそれに対応する B での観測結果を比較することにある。C2 では A 側の観測結果が早期に送られているにもかかわらず B 側の観測が未完了で比較できない。一方、C3 では B 側での観測が完了しているにもかかわらず相手側から比較すべ

き項目が送られてきていない。

これらをまとめると次の問題が指摘できる。

- 不要オペレーションの実行
 - 相手からの不要なデータに惑わされる。(2)
 - LP で候補となった不要なオペレーションを選択する。(1,3)
- 送られたデータの重要さが不明
 - 相手側から送られたデータの重要さが分からない。(2,3)
 - 相手から送られたデータのうちどれが重要か分からない。(2,3,4)
- 問題を相手に認識させられない
 - 重要なデータを送らない(4)
 - 認識に必要なオペレーションを実行依頼しない(4)

5 コントロール

この問題に対処するために、以下の機能をとりにれた。

Plan の通知： Plan 名を相手に通知して、フォーカスや実行オペレーションを推論させる。

診断状態数： 問題の認知を表す。問題の認知はフォーカスを得たことであり、問題解決の主導権を持てると考えられる。

有用度： 送ろうとしているデータが相手側でどの程度有用であるかを推論した値

重要度： CTL 部が実行すべきオペレーションを選択するとき参照する値

以下では各項目についての説明と実験の結果を述べる。

5.1 プランの通知

各 LODES エージェントは同型 (homogenous) である。その知識と推論メカニズムは同じで、ネットワーク管理用のデータと物理的 / 理論的配置が異なるだけである。したがって、選択されたプランを相手に知らせれば、そこで実行されるオペレーションの種類を推定できる。たとえば、3.2 節の問題点 (4) のように、相手に重要なデータを送っても相手側でそれに対応する属性値を求める

オペレーションが実行されていないければ、ネットワークモデルを比較できない。仮に相手側からプランが送られてきていれば、そのオペレーションが既に実行されたかある程度知ることができる。また問題 PA, PB のように、相手側で問題を認識しているかどうかも分かる。

本実験で、プランの通知とは採択されたプラン名を相手に順次知らせることとする。データ転送、相手側の推論のための JOB 依頼の前に、相手のプラン履歴を参照し、送付の必要がある場合にのみ転送する。ここでは C2 をベースにプラン通知の機能を付け加えた。表 1,2 の C4 に実験結果を示す。

PA ではプランの通知が効果的に働く。A 側で B のプランを認知することで、不要なデータ転送を防げる。この問題では B 側が B 側のホストのハード異常を調査するプランが採択されていれば、A の持っているデータは不要であり転送しなくなる。そのため不要なオペレーションの実行を防げる。ただし、実験ではタイミングによりプラン通知が遅れ、不要なデータ転送が起こることも観測された (C4 の括弧内の数値)。一方、PB では転送は多少は減少するものの、不要オペレーションの実行数は変わらない。これは第一に、重要なデータは相手側から送られるため、本質的に転送数を減少できない。第二に、特に B 側で行われる不要なオペレーションは LP からの候補であり、B 側のみの理由である。第三に A 側でも、問題認知後に B から送られてくる実行を優先することはできない。三番目の問題は、プラン通知をその名前だけに限っているからである。さらに必要な情報はそのプランを選んだ確信度である。

5.2 診断状態数

診断状態数 (Diagnostic Status Number) は各エージェントが問題解決できそうな確信度を表す。言い替えると、エージェントが問題の「型」を同定できそうな可能性を示す。診断状態数が上がると、エージェントが単独型の問題と判断すれば単独で、協調型の問題ならそれに応じて協調診断をおこなう。したがって、このエージェント全体をコントロールする形態となる。これは前節で説明した確信度付きプラン通知と同じ効果が期待できる。

診断状態数とその意味を表 3 に示す。初期状態

では問題に参加している LODES エージェント達は、どこに原因が位置するか分からないので3という値を持っている。診断状態数は CTL によって選ばれたオペレーションの結果に基づいて、診断ルールが次のオペレーションを候補に上げるとともに、診断状態数を設定する。通常の推論では2から4の間で推移する。

エージェント間のメッセージに診断状態数を付加して送り、その数値を元にオペレーションを選択する機能を加えたときの実験結果が C5³である。PA では、特に B 側において不要なオペレーションの実行が除去できる。これは、A からの不要なデータ転送は行われるものの、B 側の診断状態数が上がっているために、それらの参照が行われず、結果として C4 の場合と同じ効果がある。一方、PB では、A 側の不要なオペレーション数が減っている。これは B 側が問題を認識したあとに A 側に JOB の依頼を行った際に、A 側の LP によって提案されたオペレーションに惑わされず、速やかに実行されたことを意味する (A 側で、B 側から依頼があるまで2つのオペレーションが実行された。しかし、依頼後は A 側で選ばれたオペレーションに惑わされることなく B からの JOB が実行された。A で実行された二つは、B が判断した)。ただし、B 内で発生した不要オペレーション (OS_L) については、重要さの比較ができないため、効果が現れない。

最後の問題に対しては、各エージェント内でも診断状態数を持ち回ることが有効であろう。診断状態数が低い時に提案されたオペレーションと高いときに提案されたものとは、当然差をつけるべきである。このときの結果が C6 である。B 内で選ばれた不要なオペレーションも減少できている。

5.3 重要度

PB におけるもう一つの問題は、A から有力な情報を与えているにも関わらず、その参照が遅れることである。A が情報を送った時点では、ともに診断状態数は3である。したがって、C2 に従って、交互に選択される。もし、A が相手にフォーカスを与える可能性があれば、その参照を

³C4 と同じようにタイミングによって数値は若干かわる。ここでは10回程度の繰り返しで、最良のものを選んだ。

表 3: 診断状態数 (DS) とその意味

DS	診断状態数の意味
5	問題の型を完全に認識し、相手側単独で解く問題ではないとき
4	問題の型の推定し、それが相手側単独で解く問題ではないとき
3	中庸。特に有力な情報はない。
2	問題の型の推定し、それが相手側単独で解く問題と判断したとき
1	問題の型を完全に認識し、相手側単独で解く問題と判断したとき

優先させる仕組みが必要である。このために、重要度という数値を導入した。

重要度は各オペレーションのある局面における重要性和診断可能性を表す診断状態数を総合した値である。このために、有用度と基礎重要度という値を各オペレーションに付与する。

有用度 (Helpfulness Number, HN) とは、相手エージェントの推論に必要なと考えられるデータの転送 (DT) と JOB の依頼 (JR₂) が、表4に示すようにどの程度相手に必要であることを表す数値である。有用度は、データ転送や JOB 依頼が記述されたルールに、その必要性を表4に示すように、1 から5までの五段階で与える。これらの数値は、経験的な問題発生頻度と推論の効率性に依拠して決定する。データ転送 / JOB の要求の際に、診断状態数と有用度のペアが送られる。

有用度の利用は二つある。第一は受け手側でその問題の重要さを認識するためであり、この数に応じて参照や実行の優先度を定める。詳しくは以下で説明する。もう一つは、有用度に応じてデータ転送や JOB の依頼を制限することである。これは送り手側の制御であり、この点ではプラン通知と一部同じ効果があると思われる。さらに、プラン通知よりも通信回数少ないので、特にネットワーク状態が悪く、通信コストが大きいときに効果がある。しかし、本実験では送り手側で通信を止める制御は導入しないこととする。詳しくは [8]。

各ルールのオペレーションに、1 から5まで数値で表した基礎重要度を付加する。基礎重要度とは、そのオペレーションが属するプランを実行

表 4: 有用度 (HN) とその意味

DS	診断状態数の意味
5	知らせることで推論が早まる可能性の高いデータ。問題 PB のように相手側で認知できない問題である可能性があり、その問題が重大な影響を与えるとき、その問題を認知させるためのデータ。
4	問題 PB のように相手側で認知できない可能性があるとき、その問題を知らせるデータ。知らせることで推論が早まる可能性のあるデータ。
3	4 と同じであるが、その問題の発生の可能性が低い場合。
2	3 と同様だが、発生率がかなり低い場合
1	不要な場合。(使われていない)

するための重要性もしくは優先度を表す。これは推論過程の局所的な重要性である。

診断過程において、LP によって選ばれたオペレーションの基礎重要度とそのときの診断可能性を表した診断状態数の和を重要度と呼ぶ。静的な重要さと動的な確信度を合わせた数値と考えて良い。一方、オペレーションに他のエージェントへの JOB の依頼 (JR₁, JR₃)、他のエージェントへのデータ要求が含まれるときには、そのオペレーションの基礎重要度 (FIMP) とエージェントの診断状態数を送る。受け手では以下のように重要度 (IMP) を定める。

$$\begin{aligned} \text{if } DS_L = DS_R \\ \text{then IMP} = DS_R + FIMP \end{aligned} \quad (1)$$

$$\begin{aligned} \text{if } DS_L > DS_R \\ \text{then IMP} = DS_R + FIMP - 1 \end{aligned} \quad (2)$$

$$\begin{aligned} \text{if } DS_L < DS_R \\ \text{then IMP} = DS_R + FIMP + 1 \end{aligned} \quad (3)$$

ここで、 DS_L , DS_R はそれぞれ受け手、送り手の診断状態数である。ここで (2)(3) における -1 と $+1$ は診断状態数を重要視するためのもので、問題解決の可能性の高いエージェントの依頼を優先させている。これはほとんどの診断において、静的な重要さよりも動的な重要さもしくは問題解決の可能性を表した数値の高い方が、全体として有益なオペレーションを選択できるからである。ただし、この結果は LODES 特有のものかどうかは、今後他の研究との比較を行わなくてはならな

い。以下の実験で、診断状態数を優先させない場合も結果として示す。

次に相手から送られてきた受け手の推論のためにデータ転送された値の参照とそれに伴うオペレーションの重要度、並びに依頼された JOB の実行については、有用度 (HN) から次のように求める。

$$\begin{aligned} \text{if } HN = 5 \text{ then } IMP = 8 \\ \text{if } DS_L = DS_R \\ \text{then } IMP = DS_R + HN \end{aligned} \quad (1)$$

$$\begin{aligned} \text{if } DS_L > DS_R \text{ and } DS_L \leq 4 \\ \text{then } IMP = DS_R + HN - 2 \end{aligned} \quad (2)$$

$$\begin{aligned} \text{if } DS_L > DS_R \text{ and } DS_L < 4 \\ \text{then } IMP = DS_R + HN - 1 \end{aligned} \quad (3)$$

$$\begin{aligned} \text{if } DS_L < DS_R \text{ and } DS_R \leq 4 \\ \text{then } IMP = DS_R + HN + 2 \end{aligned} \quad (4)$$

$$\begin{aligned} \text{if } DS_L < DS_R \text{ and } DS_R < 4 \\ \text{then } IMP = DS_R + HN + 1 \end{aligned} \quad (5)$$

ここでは診断状態数をさらに重要視している。これはエージェントが診断状態数が 4 以上であれば、問題に対する何らかの対処方法を導き出せることを意味しているからであり、なるべくそのエージェントに任せの方がよい。一方、有用度は相手の観点から定めたもので真の重要さであるとは言いきれない。ただし、有用度 5 のものについては常に参照されるべきである。実際、推論に必ず必要なデータか、相手の推論過程を削減できることが多いからである。診断状態数は診断の可能性を表すだけで、推論過程の最適さを意味するものではないので、特に後者の場合を忘れてはならない。

表 12 の C7 に重要度を導入したときの実験結果を示す。問題 PA では変化は無いが、PB においては B 内で、A からの有力な情報の参照に先行して実行される不要なオペレーションが 1 つなくなる。これは始めに A 側から送られたキーとなるデータの有用度が比較的高く (4)、かつともに診断状態数 3 なので、そのデータの参照が迅速に起こるからである。この後に B は問題を認識し B の診断状態数が上がる (4 となる)。相対的に診断状態数が 3 のときに選ばれたオペレーションは後回しになり、重要な (この場合には OS_R 内の) オペレーションが選択される。一方、A では診断状態数は 3 のままで、B が問題認知した後送ら

れてきた重要度が大きくなり B の要求が優先される。

5.4 診断状態数付きプラン通知

ここでは詳しくは述べないが、診断状態数付きプラン通知も有効な手段と予想される。これは、オペレーションには何も付加せず、プラン通知にそのときの診断状態数を付加するものである。特に homogenous なエイジェントでは、プランを通知するだけで、多くの情報を得ることができる。たとえば、相手が診断状態数の高いプランを持っていれば、将来自分側に依頼されるかもしれない。また Single User Mode のように、相手もフォーカスを持っているが、自分がさらによいフォーカスを持っている場合は、単純な重要度よりも、プランにより相手の認識度を推定の方がエレガントかもしれない。

6 まとめ

本論文では、単独で診断できる問題と協調して診断する問題が混在している場合に必要コントロールについて述べた。各エイジェントが、直面している問題のタイプを推定し、それに応じて各オペレーションの重要性を決定する。ここでは、診断のプランに対する各オペレーションの静的な重要性和、診断を進めながらエイジェントが問題のタイプの認知の確信度を示す動的値の融合を最終的な重要度と定めた。また、問題のタイプの推定が正しくなかったときのために、他のエイジェントも可能な範囲で推論を進行させることもできた。これによりあらゆる問題に対し、効果的な推論を可能とするものである。実験を通して、重要な推論を妨げるような不要な作業を除去できることを確認した。

今回は、最後のプランについては十分な実験を行えなかった。一般にプランを通知することは有効な手段ではあるが、同時に通信回数が増加し、特にネットワークの問題では障害を持つかも知れないネットワークを通じて LODS エイジェントが会話を行っているため、ある局面ではプランの交信は不利な結果を招く。これを解決するために伝えるべき情報を溜めて、まとめて送る方法もあるが、タイムリー性に欠ける。今後は、これらの条件をすべて考慮してプランの有効性を検討したい。ここではプランの利用の代用として重要

度を導入した。

参考文献

- [1] L. D. Erman, F. Hayes-Roth, V. R. Lesser and D. R. Reddy. "The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty," *Comput. Survery*, Vol. 12, No.2, 1980, pp.213-253.
- [2] V. R. Lesser and D. Corkill. "The Distributed Vehicle Monitoring Testbed," *AI Magazine*, Fall, 1983 pp.15-33.
- [3] E. D. Lieberman and J. L. Woo. "SIGOP-II: A new computer program for calculating optimal signal timing patterns," *Trans. Res. Rec.*, Rep. 596, 1976.
- [4] S. Cammarata, D. McArthur and R. Steeb. "Strategies of Cooperation in Distributed Problem Solving," *Proc. of IJCAI*, 1983, pp.767-770.
- [5] V. R. Lesser and D. Corkill. "Functionally Accurate, Cooperative Distributed System," *IEEE Trans. on Syst, Man, and Cyber.*, Vol. 11, No. 1, 1980 pp.81-96.
- [6] T. Sugawara. "A Cooperative LAN diagnostic and Observation Expert System," *Proc. of IEEE International Phoenix Conf. on Comp. and Comm.*, 1990, pp. 667-674.
- [7] 菅原俊治. "LAN の診断エキスパートシステムについて," 夏のプログラミングシンポジウム報告集, 1989.
- [8] T. Sugawara. "Communication Strategy Decision by Costs," *Proc. of 10th AAAI International Workstop on Distributed Artificial Intelligence*, 1990.