分散型演繹データベースシステム：SD³とそのプロトコル†

山崎 晴明††

本稿では，分散型問題解決システムのプロトタイプとして現在開発中の，演繹検索機構をもった分散データベースシステムの概要を紹介する．次に，分散システムの編成形態を形式的に考察することで，それぞれのタイプの類別とそこで扱う処理の差異を議論する．続いて，いずれの編成形態をとる分散システムに対しても適用可能なプロトコルおよび分散処理方式を提示し，あわせて本システムにおける分散型演繹検索方式の概要を述べる．次に，演繹データベースにおける全解探索という問題に対して，システム効率と各ノードプロセッサに割り当てる知識量との関係を評価し，通信に対する処理時間の比，および割り当てられる知識量の二つが，システムの効率を決定するうえできわめて重要な因子となることを示す．

1. はじめに

パーソナルコンピュータ，オフィスコンピュータ等の普及により，これらのコンピュータ上に簡易なデータベースを構築する試みが，最近盛んに行われるようになった．しかしながら，これらのデータベースがオフィス等への応用において，真にその効果を発揮するためには，これらをネットワークによって有機的に統合する必要が生ずる．このとき発生するさまざまな技術的課題を解決するための技術として分散データベースが注目されている[1)-3)]．

一方，将来の高度で多様な応用に対処するため，データベースに推論の機構を取り込んだり[5)]，広く問題解決や意思決定支援等が可能なように，知識ベース化，エキスパートシステム化する試みも盛んに行われつつある[4)]．したがって，将来には，こうした知識ベース，エキスパートシステム群を統合，管理する技術が必須となることが予想される．このようなシステムを分散型問題解決システムと呼び，その代表的な例として分散型意思決定支援システムをあげることができる．

たとえば，ある企業のプロジェクト管理という応用を想定しよう．システム内には，各プロジェクト対応のローカルなエキスパートシステムがあり，それぞれの人員，進捗，予算等プロジェクト遂行上の種々の管理と意思決定サポートとを行っている．これに対し，新規プロジェクトの発生により，経験年数n以上の技術者m人が必要となったとする．全体への影響が最も少ないm人の選定といった問題には統合化したグローバルな意思決定が必要となる（**図1**参照）．

こうした分散型問題解決システムは，その重要性にもかかわらず，あまり考察されていない．報告された研究事例も数少ないが，このようなシステムへの試みとしては，スタンフォード大学で開発した Contract Network[6)] と CMU (Carnegie Mellon University) における，分散型 Hearsay II 等の経験をもとに Lesser 他により提唱された分散型解釈モデル (distributed interpretation model)[7)] とをあげることができる．Contract Network は，マネージャとコントラクタと呼ばれる動的に変化する役割を担う二者間でのプロトコルを規定したもので，システムの編成形態が明確で部分問題への分割が比較的容易に行える場合には適しているが，そうでない場合，制御がきわめて複雑になってしまうことが予想される．

これに対し，分散型解釈モデルでは，情報をノード間で相互交換し，その情報を自ノードへフィードバックするという形で問題解決が進行するため，明確に部分問題に分割できないようなものに対しては効果的であるが，部分問題が比較的明確に定義できるような問題に対しては，重複処理の発生の可能性がきわめて高いため，適切なアプローチとはならなくなる．

またこれらの研究事例においては，各ノードへの割り当て知識量とシステム全体のパフォーマンスといった要因についての考察があまりなされていない．

本稿では，まず2章で分散型問題解決システムのプロトタイプとして現在開発中の分散型演繹データベースシステムの概要[8)]を紹介し，次に3章でシステムの編成形態を形式的に考察することで，そのタイプの類別と扱う処理の差異を明確にする．続いて，いずれの形態の分散システムに対しても適用可能なプロトコル

† A System for Distributed Database with Deductive Search Mechanism: SD³ and Its Protocol by HARUAKI YAMAZAKI (Information Network Department, Systems Laboratory, OKI Electric Industry Co., Ltd).

†† 沖電気工業(株)総合システム研究所情報ネットワーク研究部

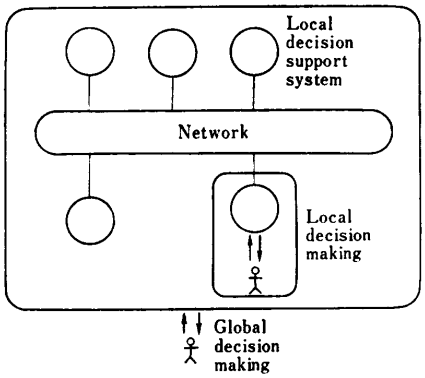

図 1 分散型意思決定支援システムの概念
Fig. 1 The concept of distributed decision support system.

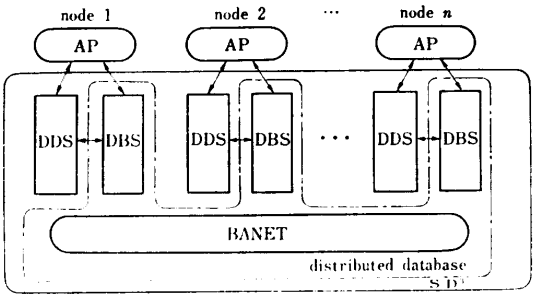

図 2 SD³ 構成概要
Fig. 2 Brief description of SD³.
AP: Application Progrrm, DDS: Data Deduction System, DBS: Data-Base Systam, BANET: Broadcast Architecture NETwork.

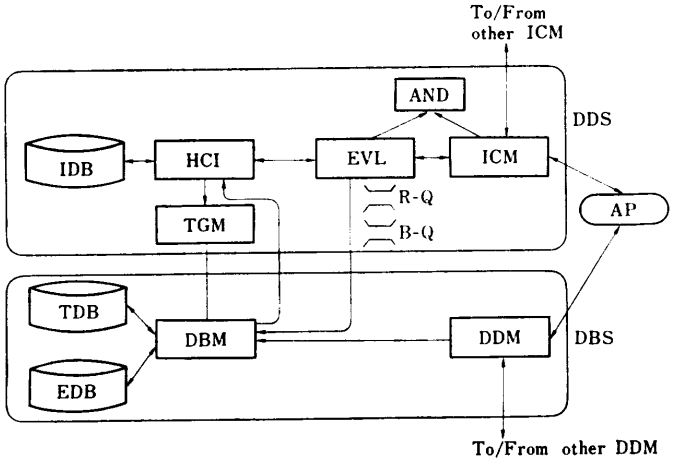

図 3 ノード内モジュール構成
Fig. 3 Modules in a node.
IDB: Intensional Data Base, TDB: Temporary Data Base, EDB: Extensional Data Base, HCI: Horn Clause Interpreter, TGM: Trigger Mechanism, DBM: Data Base Manager, EVL: Evaluator, ADN: Area Description, ICM: Inter-node Communication Manager, DDM: Distributed Database Manager, R-Q: Ready Queue, B-Q: Blocked Queue.

と探索の機構とを提示し，あわせて本システムにおける分散型演繹探索方式の概要を述べる．さらに演繹データベースにおける全解探索という問題に対して，システムの効率と各ノードに割り当てる知識量との関係を評価する．次に4章では，本システムの開発状況を報告し，今後の技術課題を示す．最後に，5章では，本システムの特長を要約する．

2. システムの概要

システムのプロトタイプとして，現在，演繹検索機構をもつ分散データベースシステム SD³ (System for Distributed Database with Deductive search mechanism) を開発中である[8]．このシステムの全体構成を**図2**に示す．図2において，AP は，具体的問題に依存した手続き部分である．各ノードは演繹推論を実行する DDS とデータベースの検索を実行する DBS とから成る．各ノードの DBS を統合することによって分散データベースシステムが構成され，さらに DDS を結合させることにより，分散型演繹データベースシステムが構成される．なお，これらのノードを結合するローカルネットワーク BANET は，同報通信をベースに構成された独自のネットワークアーキテクチャであり，コミットメント制御機能をはじめ，分散データ処理を行ううえできわめて重要な機能をネットワーク中に取り込んでいる[9),10)]．

図3は各ノードにおけるモジュールの論理構成を示したものである．図3において，ICM は分散型問題解決のプロトコルを実行し，ADN を用いて他ノードからの受信情報が自ノードに関連するか否かの判定，他ノードへの自身の導出した情報の送出等を行う．

EVL は，探索戦略を実行し，評価関数に沿った探索，全解探索のいずれも実行可能であり，また探索の深さも自由に設定できる．また自ノードで導出した情報が自ノードでは処理できない場合は，それを ICM に渡す．

IDB には，ホーン節で記述されたルールベース，関数定義等が格納されており，HCI は，EVL より与えられたゴール節をもとに演繹探索を実行する．なお探索の戦略 (e.g. depth/breadth first，評価値による枝刈り等)，および探索の深さは，EVL が DBM を介して TDB に書込みを行うことで，HCI に，指示がなされる．

PARTS

Parent	Child
a	b
a	c
a	f
b	d
b	g
⋮	
c	e

図 4 PARTS リレーション
Fig. 4 PARTS relation.

EDB には探索を実行するために必要なファクト集合が，TDB には導出された中間結果および EVL により指示された探索戦略が格納される．これらを管理するのが DBM であり，さらに DDM は分散データに対する検索，更新の同期等の処理を実行する．TGM は，探索の途中で導出された中間結果を TDB に書き込むためのトリガ機構である．

これらモジュール間の関係をより明確に示すため，部品管理において，ある部品（"a" とする）の孫部品を求める問題を例に，システムの動作を概説する．

まず，EDB には親部品と子部品の名前の対が PARTS リレーションとして格納されているものとする（**図 4** 参照）．

本システムにおいて特徴的なことは，導出の過程で用いられる中間結果に識別子を付し，さらにそれに対する述語を定義することにより，探索の戦略を指示できることにある．以下に TDB および IDB に格納される内容を示す．ただし，ここで述語 CONT $(h, \boldsymbol{x})$, TEMP $(h, \boldsymbol{x})$, C. TEMP $(h, \boldsymbol{x})$ は識別子 h をもつ中間結果の内容はリスト $\boldsymbol{x}$ であることを，述語 REL (i, j) は識別子 j をもつ中間結果が識別子 i をもつ中間結果の直接の後続結果であることを示す．また $f(\boldsymbol{x})$ はリスト $\boldsymbol{x}$ の最初の要素を取り出す関数である：

(TDB の内容)

TEMP $(h, \boldsymbol{x})$,

REL (i, j).

(IDB の内容)

Rule 1: CONT $(h, \boldsymbol{x}) \leftarrow$ TEMP $(h, \boldsymbol{x})$

Rule 2: CONT $(h, \boldsymbol{x}) \leftarrow$ REL (i, j),
CONT $(i, \boldsymbol{y})$,
PARTS $(f(\boldsymbol{y}), f(\boldsymbol{x}))$,
C. TEMP $(i, \boldsymbol{x})$.

trigger 1: [Insert to TEMP] on Rule 2,

trigger 2: [Delete from C. TEMP] on Rule 2.

ここで，C. TEMP は，述語 TEMP を真とする要素集合の補集合であり，一度中間結果として導出してしまった要素を再度導出するのを回避するために用いられる．

いま，"a" の孫部品のうちその親部品が異なっているものを二つだけ求めるには，以下の手順が実行される：

(1) EVL は，データベース REL にレコード (h_0, h_1), (h_0, h_2), (h_1, h_3), (h_2, h_4) を書き込む．これは識別子 h_0 をもつ中間結果の直接の後続結果の識別子を h_1, h_2 とし，h_1 の後続結果を h_3, h_2 のそれを h_4 とするということを意味している（探索戦略）．

また，TEMP データベースには，レコード (h_0, a) を書き込む（初期値）．

(2) EVL は HCI に対しゴール節 $\leftarrow$CONT $(h_3, \boldsymbol{x})$ を出す．

(3) HCI はルールを順次適用し，まず Rule 2 より，CONT $(h_1, (b, a))$ を求める．

(4) トリガが起動されレコード $(h_1, (b, a))$ が，TEMP に書き込まれる（中間結果の書込み）．

(5) HCI は Rule 2 を再帰的に実行しているから，再び Rule 2 により CONT $(h_3, (d, b, a))$ を求める．

(6) トリガが再び起動され，レコード $(h_3, (d, b, a))$ が，TEMP に書き込まれる．

(7) HCI は EVL に，解 $\boldsymbol{x}=(d, b, a)$ を返す．

(8) EVL は，HCI にゴール節 $\leftarrow$CONT $(h_4, \boldsymbol{x})$ を出す．

(9) 同様の処理が行われるが，Rule 2 の末尾の述語 C. TEMP により，CONT $(h_2, (b, a))$ は解から除く．

(10) 上記(3)～(7)と同様の処理が実行され解 $\boldsymbol{x}=(e, c, a)$ を返す．

このように，本システムは，木探索の問題に置き換えて，データベースの演繹検索を実行するものでありEVL が HCI に与えるゴール節は，中間結果の識別子を指定してその内容を問うという形式になっている．ここで，問われるのは中間結果の内容である必要は必ずしもなく，場合によってはその評価値であってもよいということに注意されたい．通常，評価値による探索問題においては，中間結果自体は探索戦略の決定には何も寄与しない．このため，後述の現在開発中の応用においては，評価値のみを得て次の探索戦略を決定し，中間結果は TDB に蓄積のみ行う方式をとっている．

3. 分散型演繹検索機構

前章で述べたように，本システムの各ノードは木探索を実行しているが，同時にまた，分散システム全体でも，組織的な，一つの木探索となっている必要がある．

以下では，こうした分散型の探索方式を形式的に扱うため，いくつかの用語を定義する：

（1） 中間結果が Fail もしくは Success 値のみを直接の後続中間結果としてもつ場合，これを**終端結果**もしくは**解**と呼ぶ．また先行中間結果をもたない中間結果を**初期値**もしくは**問題**と呼ぶ．

（2） システム内のあるノードに解ではない一つの中間結果が与えられたとする．その直接の後続中間結果の少なくとも一つがそのノードで導出できるとき，与えられた中間結果はそのノードにとって**展開可能** (expandable) であるという．さらに，すべての後続中間結果をそのノードで導出できるとき，その中間結果は，そのノードにとって**全展開可能** (totally expandable) といい，そうでない場合は，**半展開可能** (partially expandable) な中間結果と呼ぶ．

（3） あるノードが展開可能な中間結果の集合をそのノードの**関連域** (area of interest) と呼び，さらに全展開可能な中間結果の集合をそのノードの**達成域** (area of completion) と呼ぶ*．

3.1 分散型探索システムの類別

関連域と達成域という概念を用いて，分散システムの編成形態を類別することができる．つまり，一つは，任意の中間結果に対し，その中間結果を達成域にもつようなノードがシステム内に少なくとも一つ存在する場合で，本稿ではこれを**全編成**された (totally organized) システムと呼ぶことにする．

これに対し，与えられた中間結果を自分の関連域内にもつすべてのノードがその展開を実行して直接の後続中間結果を導出，その和集合をとれば，与えられた中間結果の直接の後続中間結果をすべて導出したことと等価となるシステムもある．このようなシステムを**半編成** (partially organized) システムと呼ぶ．

一方，各ノードの保持するルールあるいはファクト集合を与えられた問題に応じて適宜移動させなければすべての解の導出が不可能な場合もある．このようなシステムを，**未編成** (disorganized) システムと呼ぶ．

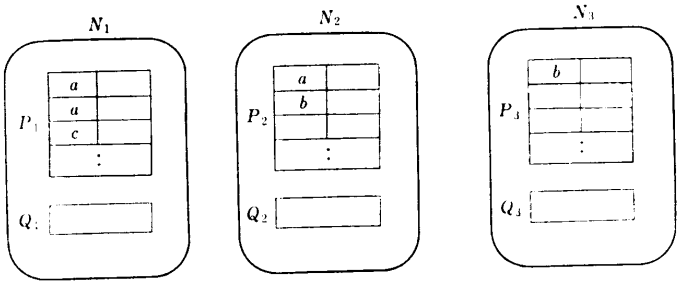

図 5 未編成システム
Fig. 5 Disorganized system.

次に分散データベースにおけるジョイン演算の例を用いて，これらの差異を論ずる：

たとえば，PおよびQをバイナリリレーションとする．いま，Pの第1カラムをある値によって制限し，Pの第2カラムとQの第1カラムとのジョインを行うキュアリ (e.g. $p(a,y) \wedge Q(y,x)$) を考える（これは深さ1の木の全解探索とみなすこともできる）．いま，PとQとをそれぞれ三つの部分 $P_1, P_2, P_3, Q_1, Q_2, Q_3$ に分割し，対 (P_i, Q_i) をノード N_i に格納したとする（**図5**）．このように配置したシステムでは，格納データの移動を伴わないで解を求めることはできない．それゆえこれは未編成システムとなる．

一方，PはそのままでQを重複して全ノードに割り当てたとする*．この場合は各ノードのローカルな join 結果をあわせれば解になるという意味で半編成システムである（**図6**）．

次に，Pを第1カラムでソートし，第1カラムが同一値をもつタプルは同一ノードに収容されるようPを再分割したとする（**図7**）．これは，一つのノードでのジョイン演算の実行により，必ず解が得られるから，全編成システムとなる．

同様に2章の例においても，リレーション PARTS を第1カラムでソートして分割配置した場合には全編成，ソートせずに分割配置した場合は半編成，rule 1 と rule 2 および PARTS をそれぞれ別ノードに配置した場合は，未編成システムとなる．

このように定義したとき Contract Network は全編成システムに最もよく適合し，また分散型解釈モデルでは未編成もしくは半編成のシステムを扱っている

---

* 実システムにおいて，中間結果が自身の関連域に在るか否かは，①判定アルゴリズム，②ディレクトリ参照，③実際の展開のいずれか，達成域の場合は①，②のいずれかにより判定される：判定が不可能な場合，達成域は存在しない．これらの判定を実現したデータ，またはプログラムをエリア記述 (ADN) と呼ぶ．

* データベースにおけるジョインの方式提案でこのようにしたアプローチをみることができる[11]．

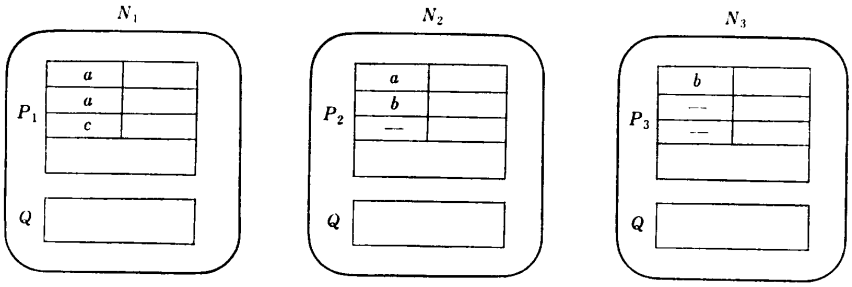

図 6 半編成システム
Fig. 6 Partially organized system.

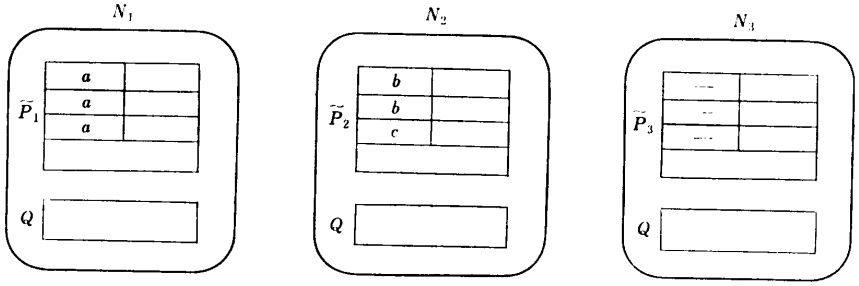

図 7 全編成システム
Fig. 7 Totally organized system.

と考えられる．また一般の分散データベースは，未編成のシステムにおいて，データを適宜転送することにより，ダイナミックに全編成システムに構成し直し，深さ1の木の全解探索を行っていると考えられる．

分散型探索システムにおいて，未編成を全編成もしくは半編成に構成し直す際にとくに問題となるのは，ファクトのみならずルールの転送の必要が生ずるということである．これはファクトの転送のみ扱うのが従来の分散データベースであったのに対し，いわば広義の分散データベースということができる．

なお全編成システムにおいては，各ノードの関連域を実際の関連域よりも縮小させ，達成域と同一の領域に定義し直してもシステム全体の機能は変わらない．このとき，関連域，達成域あわせてたんにエリアと呼ぶ．

3.2 プロトコル概要

本稿では，分散データベースプロトコルを，ルールやファクトの転送により未編成システムを半編成もしくは全編成システムに動的に構成し直すものであると定義する．これに対し，分散型問題解決プロトコルとは，分散データベースプロトコルの動的実行により，またはデータベースの静的配置により，全編成もしくは半編成となったシステムのうえで，組織的に全体問題に対する解を導出するためのプロトコルと位置づける．なお，本システムの分散型問題解決プロトコルは全編成，半編成どちらのシステムに対しても同一のプロトコルの適用が可能である．以下にプロトコルの概要を述べる．

なお，R-Q, B-Q は中間結果をリンクづけするためのキューであり，R-Q には展開の対象となる中間結果が，B-Q にはその後の展開を他ノードに委託しようとしている中間結果がリンクづけされる．

① あるノード N_i に問題 s_0 が与えられる．

② N_i は s_0 から出発し，探索を開始する．

③ 導出した中間結果 s_i が N_i の達成域を越えた場合，N_i は協調メッセージ (CO-OP message) により，s_i を全ノードにブロードカストする．ただし，s_i が N_i の関連域にある間は N_i は s_i に対する展開を続行する．s_i が関連域を越えてしまった場合は，それを B-Q におく．

④ N_i は自身の関連域に解以外の中間結果が，存在しなくなるまで，R-Q 内にある中間結果の展開を続行する．

⑤ s_i を受信したノード N_j は，それが N_j の関連域にあれば，優先度 P_j を付加して競合メッセージ (comp-message) を全ノードにブロードカストする．なお付加する優先度はシステムが全編成か半編成かによって異なる：

全編成の場合，自ノードの R-Q にある中間結果の数 r_j とノード番号 N_j との対とする．半編成の場

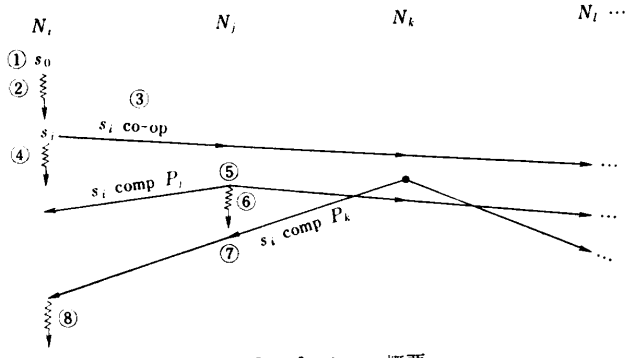

図 8 プロトコル概要
Fig. 8 The outline of protocol.

合，s_i が達成域にあれば 0，そうでなければ 1 とする．

⑥ 競合メッセージを送出したノードはただちに s_i を R-Q におき，展開の対象とする．

⑦ 他ノード N_k からの s_i に対する競合メッセージを受信した N_j は，優先度を比較し，自身のほうが低ければ R-Q より，s_i およびその後続中間結果を取り除く．優先度が同じか，自身のほうが高ければ処理を続行する．ただし優先度は，全編成の場合 r_j と r_k の少ないほうが優先し，もし同じであればノード番号が比較される．半編成の場合は 0 が 1 に優先する．

⑧ 一つでも他ノードからの s_i に対する競合メッセージを受信した N_i は，s_i を B-Q より取り除く．もし s_i およびその後続中間結果が R-Q にある場合は，⑦と同様の処理を行う．

図 8 にプロトコル概要を示す．

3.3 分散型探索システムの評価

本節では評価対象として，全編成システムにおける全解探索問題をとりあげる．このとき，システムの実行効率にとくに影響を与える要因として重要なものは，各ノードの保持するエリアの大きさである．つまり，エリアを大きくしすぎると並列実行による効果がほとんど得られない（問題が単一ノードに閉じてしまう）し，またエリアを小さくしすぎると通信オーバヘッドがきわめて大となってしまうことが予想される．

この最適なエリアの大きさを，本節では展開の結果生ずる直接の後続中間結果が，もとの中間結果と同一エリアに入る確率という観点から論ずる．

（1） 前　　提

① 一つの中間結果を展開して直接の後続中間結果を一つ生成するのに要する時間を一定値 u sec とする．

② 一つの中間結果を他ノードに転送するのに要するデータのビット長を B，通信チャネルの実効スループットを T bit/sec とする．$r=B/T$ は，一つの中間結果を別ノードに転送するため通信チャネルを専有する時間となる．したがって通信・処理比 C は，$C=r/u$ となる．

③ 中間結果の展開処理と送受信処理は，完全に並列に実行されるが，通信チャネルは単一であり，すべてのノードが共有する．

④ 確率変数 $\boldsymbol{Z}$ を，一つの中間結果から発生するすべての直接の後続中間結果の総数とする．もし与えられた問題が解をもつのであれば $E(\boldsymbol{Z})=\rho<1$ である．

⑤ 確率変数 $\boldsymbol{X}, \boldsymbol{Y}$ をそれぞれ，先行中間結果と同一エリアに属する直接の後続中間結果の総数，および異なるエリアに属する中間結果の総数とする．

⑥ $E(\boldsymbol{X})=\lambda, E(\boldsymbol{Y})=\mu$ とし，$\sigma=\lambda/(\lambda+\mu)=\lambda/\rho$ とおく．σ を探索のローカリティと呼ぶ．これは，直接の後続中間結果が，先行中間結果と同一エリアに属する確率である．

⑦ 与えられた問題を第 0 世代，それから直接生じた中間結果の集合を第 1 世代，…と呼び，各世代集合の要素の数の分布を $\boldsymbol{Z}_0, \boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n, \cdots$ で表す．ただし，$\boldsymbol{Z}_0=1, \boldsymbol{Z}_1=\boldsymbol{Z}$ である．

（2） 探索ローカリティの最適値

$\varphi(z)$ を $\boldsymbol{Z}$ の分布の母関数，$\varphi_X(z)$, $\varphi_Y(z)$ をそれぞれ $\boldsymbol{X}, \boldsymbol{Y}$ の分布の母関数とする．いま，$\boldsymbol{Z}_n$ の分布の母関数を $\varphi^{(n)}(z)$ で表すと，変数 $\boldsymbol{Z}_0, \boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n, \cdots$ は分枝過程 (branching process)[12] を構成するから，

$$\varphi^{(n)}(z)=\varphi(\varphi^{(n-1)}(z))$$

が成り立つ．したがって，第 n 世代集合の要素の数の平均 Z_n は；

$$Z_n=\left[\frac{d\varphi^{(n)}(z)}{dz}\right]_{z=1}=\varphi'(\varphi^{(n-1)}(1))\cdot\varphi'^{(n-1)}(1).$$

$\varphi'(1)=\rho$ であるから $z_n=\rho^n$ となる．したがって，第 0 世代から解にいたるまでに生じた要素の総数は $z=\sum z_n=1/(1-\rho)$ となる．したがって単一ノードで探索を行ったとき要する平均時間 T_{max} は，

$$T_{max}=u/(1-\rho) \qquad (1)$$

である．

一方，$\boldsymbol{Z}=\boldsymbol{X}+\boldsymbol{Y}$ であり，また展開された中間結果の属するエリアが先行中間結果のそれとは異なる場合，1 回，通信を伴うから，$n-1$ から n 世代を生じたとき要する通信回数 $\boldsymbol{Y}_n$ は，変数 $\boldsymbol{Y}$ の独立な $\boldsymbol{Z}_{n-1}$ 回の和となる．したがって $\boldsymbol{Y}_n$ の分布の母関数は φ^{n-1}

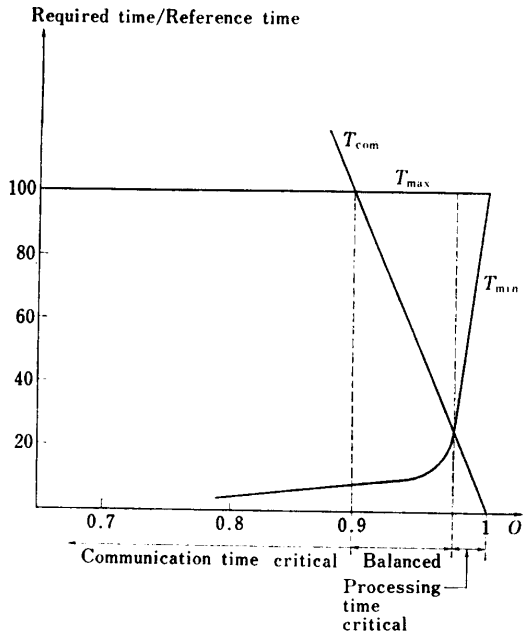

図 9 $C=10, \rho=0.99$ の場合の $T_{max}, T_{min}, T_{com}$
Fig. 9 The curves of $T_{max}, T_{min}, T_{com}$ where $C=10$, $\rho=0.99$.

$(\varphi_Y(z))$ で与えられる[12]. ゆえに $\boldsymbol{Y}_n$ の平均値 y_n は (2)式で与えられ

$$y_n=\left[\frac{d\varphi^{(n-1)}(\varphi_Y(z))}{dz}\right]_{z=1}$$
$$=\varphi'^{(n-1)}(\varphi_Y(1))\cdot\varphi'_Y(1), \qquad (2)$$

探索が終結するまでに要した通信時間の平均値 T_{com} は，(3)式で与えられる：

$$T_{com}=c\cdot u\cdot(\sum y_n)=c\cdot u\cdot\mu/(1-\rho). \qquad (3)$$

一方，あるエリアに属する一つの中間結果が同一エリアに後続中間結果を生成しなくなるまでの平均時間を T_{min} とすると，ρ を λ で置き換えたときの T_{max} と同様にして，$T_{min}=u/(1-\lambda)$ を得る．これは探索に要する最少時間の平均値を与えているとみなすことができる．

$\sigma=\lambda/\rho$ から，(4)式，(5)式を得る：

$$T_{com}=c\cdot u\cdot\rho(1-\sigma)/(1-\rho) \qquad (4)$$
$$T_{min}=u/(1-\rho\sigma) \qquad (5)$$

図9 に $c=10, \rho=0.99$ とした場合の $T_{max}, T_{min}, T_{com}$ を示す．σ の最適値は通信時間と処理時間の等しいところ，図9の例においては，0.895 から 0.975 までの間にあり $\sigma>0.975$ で処理ネック，$\sigma<0.895$ で通信ネックとなることがわかる．なお，これはノード総数，エリアの重複の程度に無関係で ρ および c の値によってのみ定まる不変的な性質であることに注意されたい．

一般に最適解 σ_{opt} は，不等式 $\sigma_0\leq\sigma_{opt}\leq\sigma_1$ を満たす．ただし，σ_0 は方程式 $T_{max}=T_{com}$ の σ についての解，σ_1 は $T_{min}=T_{com}$ の σ についての解である．(4)式および(5)式と，$\sigma\leq1$ とから，

$$\sigma_1=(1+\rho(1-\rho)^2+4(1-\rho)/c)^{1/2})/2\rho \qquad (6)$$

を得る．

同様に $\sigma_0=1-1/(c\rho)$ となる．$0\leq\sigma$ だから，$c\rho<1$ の場合，σ_0 は定義されない (i.e. どのような σ の値に対しても通信ネックとはならない)．したがって σ_{opt} の範囲は，一般に(7)式で表される：

$$\max(0,(1-(c\rho)^{-1})\leq\sigma_{opt} \qquad (7)$$
$$\leq(1+\rho-((1-\rho)^2+4(1-\rho)/c)^{1/2})/2\rho$$

4. 開発の現状

現在，SD³ システムの応用として略地図発生システムを開発中である[13]．これは，ある特定地域における建物，交差点およびそれらの間の接続関係と距離といった地図情報を関係データベースとして表現し，それらを適宜重複を許して各ノードに格納した EDB と，完全に重複させた IDB とから，与えられた2点間のパスを略地図として出力するという一種のプラン生成システムである．パスの導出は評価値，全解探索のいずれも可能である．現システムでは，初期状態で全編成となるように EDB, IDB の分割を行っている．また各ノードのエリア記述は，2次元の領域として座標で表現される．

システムの構成は，ノードとなる5台のパーソナルコンピュータと3台のネットワークアダプタ装置から成っている．なお各モジュールは，一部デバグ中のものを除き，コーディングもしくはプログラム設計中である．

5. 結　　論

本稿では，演繹検索機構をもった分散データベースシステム SD³ の概要紹介と問題解決のためのプロトコルの提案を行った．本システムの特徴を要約すると以下のようになる．

(1) 問題解決のための制御構造を，HCI から分離した．このため，多くの演繹検索を一般的な木の探索問題として表現することができ，柔軟に探索戦略を変更することが可能となった．

(2) TDB に中間結果を生成し，それをデータベースの管理下に置いたため，膨大な中間結果への迅速なアクセス，重複した探索の回避等が可能となっ

た.

(3) キュアリ機能を備えた分散データベースとしても，動作するため，APに柔軟なインタフェースを提供できた.

(4) 全編成，半編成いずれのシステムに対しても適用可能な分散型問題解決プロトコルの導入を行った.

なお現システムでは静的に全編成となるようEDB，IDBの分割を行った．今後は半編成や未編成システムとなる応用についても開発を予定している．とくに未編成システムの場合，ファクトデータのみならず，ルールの転送をも含めた広義の分散データベースプロトコルの設計や，分散キュアリの実行戦略の検討といった問題が今後の主要な研究課題として残されている.

参考文献

1) Rothnie, J. B. and Goodman, N.: A Survey of Research and Dovelopment in Distributed DataBase Management, Proc. of Int. Conf., 3rd VLDB, pp. 48-62 (1977).
2) Rothnie, J. B. et al.: Introduction to a System for Distributed Database (SDD-1), *ACM Trans. on Database System*, Vol. 5, No. 1, pp. 1-17 (1980).
3) 山崎：分散データベース—更新トランザクションの実行と制御に関する記号演算，情報処理学会論文誌，Vol. 23, No. 1, pp. 35-42 (1982).
4) 大須賀：知識ベース技術の展望，情報処理，Vol. 23, No. 10, pp. 967-974 (1982).
5) Minker, J.: An Experimental Relational Database System Based on Logic, in *Logic and Databases*, pp. 107-147, Plenum Press, San Francisco (1978).
6) Smith, R.: The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver, *IEEE, Trans. Comput.*, Vol. C-29, No. 12, pp. 1104-1113 (1980).
7) Lesser, V. and Erman, L.: Distributed Interpretation: A Model and Experiment, *IEEE Trans. Comput.*, Vol. 29, No. 12, pp. 1144-1163 (1980).
8) 吉田他：演繹機構を持つデータベースシステムの一考察，情報処理学会第28回全国大会，pp. 733-734 (1984).
9) 岸田他：分散処理向きローカルネットワークについて，情報処理学会，分散処理システム研資，14-4 (1982).
10) Yamazaki, H. et al.: A Proposal for Broadcast Architecture Network (BANET), Proc. of 6th ICCC, pp. 115-120 (1982).
11) Valduriez, et al.: Multiprocessor Join Algorithms of Relations, in *Databases: Improving Usability and Responsiveness*, pp. 219-236 Academic Press, New York (1982).
12) Feller, W.: *An Introduction to Probability Theory and Its Application*, Vol. 1, Wiley, New York (1950).
13) 古田他：演繹機構を持つデータベースによる略地図発生システム，情報処理学会第28回全国大会，pp. 735-736 (1984).

(昭和59年5月21日受付)
(昭和59年9月20日採録)