

複数属性の準結合を用いる分散データベース質問処理法

李 紅

(株) リコー ソフトウェア事業部

佐藤 洋

電気通信大学 情報工学科

分散データベースの質問処理に複数属性による準結合と結合をレデューサーに入れる方法を取り入れたアルゴリズムを提案する。そのために、関係の n 次の次数という新しい概念が導入され、その定量化も行なわれる。このような研究方向は初めての試みであり、意義があるものと思われる。複数属性による関係間の結合演算は集中型データベースにあまり多くないが、分散データベースの質問ではしばしば現れる。

Query Processing Using Multi-Attributes Semijoin in Distributed Database

Hong LI

Software Division. RICOH COMPANY, LTD.

1-17, Koishikawa 1-Chome, Tomin-Bldg

Bunkyo-ku, Tokyo, 112, Japan

E-mail: lihong@src.ricoh.co.jp

Hiroshi SATO

Department of Computer Science

University of Electro-Communications,

1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182 Japan

We propose a query processing algorithm for distributed database management systems. The algorithm inputs joints and semijoints of multi-attributes to reducers. In order to realize it, we introduced a new concept called n -degree for relations and gave it some quantitative analysis. We consider these new approaches of research in this field has significant value.

1 はじめに

分散データベースは従来の集中データベースと比べてその挙動が大きく変わっている。質問処理だけを対象としても、通信回線上でのデータ転送に応じて、質問処理の最適化も違う手法を用いざるを得ない。これに関する研究は最近増加しつつあり、少なくない成果が得られている[1]～[11]。

しかし、いままでの分散データベースの質問処理の最適化に関する研究は、分散データベースに多く存在する複数属性で結合する質問を考えていない傾向があるが、これは現実と掛け離れている。本論文ではこの問題および結合を最適化の手段とする分散質問処理の最適化について論ずる。同時に、上記の問題を解決する関係の n 次次数の概念と質問処理アルゴリズムを提案する。このアルゴリズムによって非常に小さい転送コストが得られることも後で実証する。

もう少し具体的に説明する。まず、複数属性による結合について述べる。次の結合に注目しよう。

関係 $R_1(a, b, c, \dots)$ と関係 $R_2(a, b, c, \dots)$ と属性 a, b, c について結合をする

これは十分に研究されていない複数属性による結合の質問である。このような質問は集中データベースにはあまり現れないものであって、分散データベースに多く現れるものと考えてもよいだろう。集中の関係型データベースの場合は異なる関係が同じ複数の属性を持つと、データが冗長になりやすいし、処理速度も遅くなりがちで、デメリットが多い。また、実際に運用するときデータベースの首尾一貫性、障害回復、データベースの保守、耐故障性などの面にも問題が出てくるであろう。分散データベースの場合は共通の属性をもつ関係を異なる場所に置くことはしばしば起こり、極端な例でいうと、水平に分割された関係はその属性が皆おなじである。従って単一属性の結合のみを最適化の対象とするモデルが不完全だと考えられる。この問題を正面から取り扱う研究はほとんどないようである。

次に、結合と準結合からレデューサー(reducer)を求める方法について述べる。通常、分散データベースの質問処理は三つのステップからなっている。(a)局所処理、(b)データ転送による関係サイズの減少、(c)結果を求めるノードへのデータ転送と演算である。ステップ(b)において、準結合を用いる研究はたくさん行なわれている。最近では、この段階で結合を導入する研究も見られる[11]。結合をレデューサーに入れることは準結合で得られない効果がある。本論文は複数結合属性による準結合の角度からこの問題についても議論したい。

本論文ではこれらの問題を考慮に入れ、分散質問処理の最適化に対して、新しい考え方と有効な処理アルゴリズムを提案する。もっとも重要なことは関係の次数の概念とその定量化及びこれをベースとした複数属性による準結合である。これは他の論文にない手法であって、その結果をいくつかの定理にまとめておいた。さらに、提案された方法の有効性を他の方法と比べて実証する。

本論文の構成は次の通りである。

第2章は分散データベースのモデルと関係の n 次次数の概念で、第3章は関係の次数と複数属性の準結合に関する定理、第4章は質問処理のアルゴリズムと例題、第5章は考察、第6章はまとめである。

2 分散データベースのモデルと関係の n 次次数の概念

2.1 分散データベースの質問処理のモデルと記述法

本論文で取り扱う分散質問処理の環境は遠隔地に分散されたコンピュータネットワークである。そのため、質問処理のコストは通信にかかるコストのみとする。コストは転送するデータの量の線型関数とし、次の式を用いる。

$$C = C_0 + \beta X \quad (1)$$

C_0 は通信設定のコスト、 β は単位データを転送するコスト、 X は転送されるデータの量である。簡単のため、 $\beta = 1$ とする。

質問の対象となるデータベースは分散された関係データベースである。質問処理はまずネットワークの各ノードで行なわれるとする。したがって、扱う関係がそれぞれ異なるノードに格納されているとする。同じデータが複数

存在するときは、事前に一つ選ばれたとする。データは関係上に均一に分布されているとし、関係の属性は統計的に互いに独立とする。

以上の仮定は多くの論文で見られる。しかし、多くの論文では結合する属性は一つであることと仮定しているか、または、一般な質問であると仮定するが、最適化の課程において単一属性の結合しか考慮にいていない。このような仮定は実際に分散データベースの質問処理の多くに対して最適化の程度を妨げている。すなわち、分散データベース環境に多く現れる複数属性に関する結合が含まれていない。

我々は複数属性による結合を含む一般な質問を最適化の対象とする。

本論文では関係を U, R 等で表わし、属性を $a, b, c, \dots$ で表わす。 $R(a, b, c)$ は属性 a と b と c を持つ関係を表わすことになる。 $R(a)$ は関係を a に射影した関係とする。 $R(a, b, c)$ のタプル数は t または T で記述する。 $R(a)$ のタプル数は t_a または $t_{R(a)}$ で表わす。 D_a は属性 a の領域の大きさを表わす。 $R(a)$ の選択度は p_a または $p_{R(a)}$ で、 $p = t_a/D_a$ である。

また以下のような記号を使う。

$R(a)$: 関係 $R(a)$, または, そのタプル数
 $|R|$: 関係 R のサイズ
 t_{ab} : 属性 a と b からなる関係 $R(a, b)$ のタプル数
 $R_1 \rightarrow R_2$: 関係 R_1 を関係 R_2 に転送し, R_2 のノードで準結合する

2.2 複数属性を持つ関係の選択度と複数属性による準結合

従来、関係 $R(a, b)$ の選択度を論じるとき、単一属性に射影した関係 $R(a)$ ($R(b)$) の選択度 $p_a = t_a/D_a$ ($p_b = t_b/D_b$) しか取り扱われていない。 t_{ab} は関係 $R(a, b)$ のタプル数とすると、次のような不等式が成り立つ。

$$t_{ab} \leq t_a \times t_b \quad (2)$$

上記の式の両端を $D_a \times D_b$ で割ると、次の式が成り立つ。

$$\frac{t_{ab}}{D_a \times D_b} \leq \frac{t_a \times t_b}{D_a \times D_b} = p_a \times p_b \quad (3)$$

複数属性を持つ関係 $R(a, b)$ の選択度 p_{ab} は上記式の左端によって定義する。

$$p_{ab} = t_{ab}/(D_a \times D_b) \quad (4)$$

同様にタプル t 個を持つ n 属性関係 $R(a_1, a_2, \dots, a_n)$ の選択度 p は

$$p = t/(D_{a_1} \times D_{a_2} \times \dots \times D_{a_n}) \quad (5)$$

と定義する。

付録で示したように複数属性を準結合するとき、このように定義した複数属性の選択度を用いる方法は単一属性のときと同様に合理性を持っている。

したがって、タプル数 t を持つ関係 R が選択度 p の複数 (単一) 属性関係によって準結合された場合、準結合された後の関係のタプル数 t' は

$$t' = t \times p \quad (6)$$

となる。結合されない属性 b のタプル数の期待値 $R'(b)$ は次の式で計算する。この式は[6]で述べた式の近似式である。

$$R'(b) = \min\{t \times p, t_{R(b)}\} \quad (7)$$

式 (2) は次のようになっていることが多い。

$$t_{ab} \ll t_a \times t_b \quad (8)$$

したがって、次式が成り立つ場合が多い。

$$p_{ab} \ll p_a \times p_b \quad (9)$$

この式の左辺は複数属性(a, b)で準結合する場合の選択度であり、右辺は属性 a と属性 b を別々で準結合する場合の選択度である。複数属性による準結合のほうがそれぞれの単一属性による準結合より結合を受ける関係をより小さくすることを示している。属性の数が多くなればレデューサーとしての効率も大きくなる。しかし、複数属性においてはその中の単一属性に注目するとデータの重複が存在する。

2.3 関係の n 次次数の概念

李, 佐藤 [8] では、関係を 1 次か 2 次かについて定性的に論じたが、ここで、もっとも一般的なケースに対して関係の次数を定量的に定義する。

【定義】関係 $R(a_1, a_2, \dots, a_n)$ のタプル数を T とし、 $R(a_i)$ ($i = 1, 2, \dots, n$) のタプル数を t_i とする。関係 R に対して整数 K_i ($i = 1, 2, \dots, n$) を次のように計算する。

$$\begin{aligned} K_1 &= \min\{t_1, t_2, \dots, t_n\} \\ K_2 &= \min\{t_1 \times t_2, t_1 \times t_3, \dots, t_{n-1} \times t_n\} \\ &\dots \dots \\ K_n &= t_1 \times t_2 \times \dots \times t_n \end{aligned} \quad (10)$$

したがって

$$K_1 \leq K_2 \leq \dots \leq K_n \quad (11)$$

$$K_1 \leq T \leq K_n \quad (12)$$

が成り立つ。

$$K_1 \leq K_2 \leq \dots \leq K_m \leq T < K_{m+1} \leq \dots \leq K_n \quad (13)$$

のとき、関係 R の次数を m と定義し、次のように書くことにする。

$$d(R) = m.$$

■

定義にしたがって、次の関係 $R(a, b, c)$ は 1 次関係である。

$R(a, b, c)$	T	$R(a)$	$R(b)$	$R(c)$
tuple	100	80	100	60

次の関係 $R(a, b, c)$ は 2 次関係である。

$R(a, b, c)$	T	$R(a)$	$R(b)$	$R(c)$	$R(a, b)$	$R(a, c)$	$R(b, c)$
tuple	5000	80	100	60	4000	3000	1000

関係の次数を導入することによって、明らかにされた分散質問の性質などについては次章で述べる。

3 関係の次数と複数属性の準結合に関する定理

前章では関係の次数の定義を与えた。この定義を定性的に解説すると、一般的には、関係の各属性のタプル数が関係のタプル数と多く違わなければその関係は1次関係であり、関係のタプル数が関係内の二つの単一属性のタプル数の積とほぼ同等なら、その関係は2次の関係である。同様に、属性 n 個の関係の次数が n のとき、その関係のタプル数は全ての単一属性のタプルの積に近いであろう。

複数の結合属性を持つ関係を他のノードに転送するときには、次の3つのことに注意しなければならない。第1に転送されるデータの量、いわゆるコスト。第2に転送されるデータによって、先方の関係はサイズがどのくらい減るか、これはベニフィット(benefit)と呼ばれ、転送されるデータの選択度と関係する。第3に転送されるデータの属性間の関連情報である。関係を属性に射影してから転送する場合は、射影によって、他属性のデータとの間の対応情報が失われることがあるが、重複するデータの量が減ることになる。選択度も射影するたびに大幅に増える。

定義により、明らかに n 個の属性を持つ関係 R はその次数 $d(R)$ が 1 と n の間である。

【定理1】 関係 $R(a_1, a_2, \dots, a_n)$ の任意の射影 R' の次数は R の次数より増えることはない。

【証明】 関係 R のタプル数を T 、関係 R' のタプル数を T' とすると、次式が成り立つ。

$$T' \leq T$$

関係 R の次数を計算するための整数を K とし、射影された関係 R' の次数を計算するための整数を K' とすると、定義によって、

$$K'_i \geq K_i \quad (1 \leq i \leq R' \text{の属性数})$$

が成り立つ。上記の両式によって、 R' の次数は R の次数より増えないことが分かった。■

実際に、射影した関係 R' の次数はかなり小さくなることが多い。

【系1】 1次関係の任意の射影は次数が1次である。

これから準結合を取るときの関係の次数について考察を行う。

【補助定理】 準結合した関係の非結合属性のタプル数の期待値を式(7)を用いて計算するものとすれば、関係 $R(a_1, a_2, \dots, a_n)$ は単一属性関係 $R_1(a_1)$ によって準結合を取っても次数が増えない。

【証明】 関係 R のタプル数を T 、関係 $R_1(a_1)$ の選択度を p 、準結合したあとの関係を R' 、そのタプル数を T' とする。また、 R の各属性のタプル数を t_i ($i = 1, 2, \dots, n$) とし、 R' の各属性のタプル数を t'_i ($i = 1, 2, \dots, n$) とする。そのとき、

$$T' = T \times p \quad (14)$$

となる。これから次式が成り立つことを証明する。

$$T' < K'_{m+1} \quad (15)$$

便宜のため、 $t_2 \leq t_3 \leq \dots \leq t_n$ とする。そのとき、

$$K'_{m+1} = \min \{ t_1 \times p \times \min\{T', t_2\} \times \min\{T', t_3\} \times \dots \times \min\{T', t_{m+1}\}, \min\{T', t_2\} \times \min\{T', t_3\} \times \dots \times \min\{T', t_{m+2}\} \} \quad (16)$$

が成り立つ。

T' が式(16)の右辺の計算結果に現れるなら、式(15)が成り立つ。そうでないとき、式(16)は次式になる。

$$K'_{m+1} = \min\{t_1 \times p \times t_2 \times t_3 \times \dots \times t_{m+1}, t_2 \times t_3 \times \dots \times t_{m+2}\} \quad (17)$$

式(17)の右辺において、両値のどちらが小さくても、 K_{m+1} の定義と式(14)を考えると、 $T' < K'_{m+1}$ が成り立つ。

すなわち、単一属性の準結合を受けた関係の次数が大きくなることを証明した。■

【定理2】 準結合した関係の非結合属性のタプル数の期待値を式(7)を用いて計算するものとすれば、関係 $R(a_1, a_2, \dots, a_n)$ は他の関係によって準結合を取っても次数が増えない。

【証明】 付録の式(25)と補助定理の証明手順により、定理が成り立つことが証明できる。■

【系2】 選択度 p の関係によって準結合されるとき、 p が十分小さければ、準結合を受ける関係は準結合したあとに1次関係にも成り得る。

【証明】 式(16)で $T' \leq t_2$ が成り立つように p を選べば系2が成り立つ。■

定理1と定理2では関係が射影されること、または、準結合を受けることで次数が小さくなる（大きくはならない）ことを示した。系2では準結合するときの選択度が小さければ準結合をしたあとの関係の次数も小さくなることを示した。

関係の次数が小さいとき、関係内でデータの重複度は少ない。例えば、1次関係では任意の二つの属性に射影すれば、そのデータは重複しない。一般的に n 属性の関係の次数が $m (m > 1)$ のとき、その関係内でタプル数のもっとも小さい単一属性はデータが極めて大きい回数重複する。定理1により、このような関係は部分属性に射影することによって次数を減らすことができるし、重複度も減らすことができる。一般に許される重複度は関係の次数と関係のサイズの両方に関係する。

関係の次数だけを考えると、ここで単に $m > n/2$ のときは高次関係と呼び、そうでないときは低次関係と呼ぶことにする。すなわち関係の各単一属性のタプル数を昇順に並べているときを考えると、その列の中間にある単一属性のデータが重複しているならばその関係は高次関係となる。

関係の次数の定義は関係の全体のタプル数に緊密に依存している。同じ属性を持つ関係の全体のタプル数が小さいとき、その次数も通常小さい。複数属性による準結合をすることによって、準結合された関係のタプル数は非常に小さくなり、これと同時にその関係の次数も小さくなる。この現象は準結合する複数属性の数に關係する。複数属性の数が多ければ多いほど結合された関係はタプル数と次数が小さくなる。

関係の次数は関係をどのように射影して転送するかを判断する重要な因子になっている。関係は次数が高い程データが多く重複して、射影された関係の選択度が小さい。このような高次の関係を部分属性に分割すると、より低い転送コストが得られる。

4 質問処理のアルゴリズムと例題

前章で関係の次数の性質について述べたが、本章では関係の次数の性質を利用した質問処理アルゴリズムについて説明する。

関係が準結合されることを考える。準結合されてから他のノードに転送するほうが準結合されず直接に他のノードに転送するよりコストが低い場合、その準結合をメリットのある準結合と呼ぶ。

次のアルゴリズムは全回線使用時間(total time)を最適化の対象とするものである。

1. 各ノードで局所処理をする。すなわち、射影と選択演算及びノード内での結合を行う。
2. 局所処理した関係を次数ごとにわけて、 i 次の関係からなる集合を G_i とする。次数のもっとも小さい集合を G_1 と書く。
3. G_1 を除くすべての関係から、サイズの最も小さい異なる単一属性を一つずつ抽出する。抽出された単一属性関係を G_1 に付加する。

4. 関係集合 G_1 から順次に度数のもっとも高い関係集合まで次のような操作を繰り返して行う。

- まず、集合内で関係をサイズの昇順にソートする。
- 集合内でサイズの小さい関係から順次にメリットのある準結合を求め、準結合をとり、これを転送列に加える。他の関係と準結合を行ってもメリットがないとき、その関係は集合から削除される。ただし、他の関係と共通の属性を持たない関係は削除しない。
- 準結合した結果の関係は高次関係なら、それを適切に低次関係に分けて、次にサイズの大きい関係に対して準結合をとる。この操作はこの集合のもっともサイズの大きい関係まで行なわれる。
- 上記の操作によって得られた結果は複数の関係もありえる。これらの関係を次に高い次数の関係集合に追加する。最も次数の高い関係集合の場合はその結果の関係を質問を求めるノードに転送する。

5. 最後に結合されていない関係を直接に質問を求めるノードに転送するか、または、ステップ4で得られた結果を適切に射影して準結合をしてから質問を求めるノードに転送するかを選択し、必要に応じて準結合を取り、質問を求めるノードに転送する。質問は質問を求めるノードで計算される。

いくつかのところで並列化を図れば、このアルゴリズムは応答時間(response time)を小さくするアルゴリズムに容易に直すことができる。

ここで例を二つ示す。

【例1】

関係	T	属性 a	属性 b	属性 c	属性(a, b)	属性(a, c)	属性(b, c)	次数	高さ
$R_1(a, b)$	80(0.008)	40(0.4)	80(0.8)	-	-	-	-	1	低次
$R_2(a, b, c)$	3000(0.003)	50(0.5)	40(0.4)	80(0.8)	1000(0.1)	1200(0.12)	1500(0.15)	2	高次
$R_3(a, b, c)$	75000(0.075)	30(0.3)	50(0.5)	50(0.5)	900(0.09)	1200(0.12)	800(0.08)	3	高次

$D_a = D_b = D_c = 100$ とする。各属性の長さは1とする。括弧の中は選択度を示す。

表1. 例題の分散データベース (一)

分散データベースは関係 $R_1(a, b)$ と $R_2(a, b, c)$ と $R_3(a, b, c)$ から構成されている。関係の次数と次数の高さも表に示している。この分散データベースに対して R_1 と R_2 と R_3 と属性(a, b, c) について結合する質問が現れるとき、その処理手順はアルゴリズムに沿って求める。

通信設定のコスト $C_0 = 5$ とする。

アルゴリズムのステップ2によって、次数ごとの関係集合は次の通りである。

$$G_1 = \{R_1\}, \quad G_2 = \{R_2\}, \quad G_3 = \{R_3\}$$

アルゴリズムのステップ3にしたがって、関係 R_2 と R_3 から単一属性関係 $R_2(b)$ と $R_3(a)$ と $R_3(c)$ を抽出し、それを関係集合 G_1 に追加する。 G_1 は次のようになる。

$$G'_1 = \{R_3(a), R_2(b), R_3(c), R_1\}$$

この関係集合において、アルゴリズムのステップ4により、次の転送が選ばれる。

$$R_3(a) \longrightarrow R_1$$

$R_2(b)$ を使ってメリットのある準結合が得られなかったため、 $R_2(b)$ を G_2 に付加しない。 $R_3(a)$ と R_1 で結合した結果 $R'_1(a, b)$ と $R_3(c)$ を G_2 に追加する。 R'_1 のタプル数は $80 \times 0.3 = 24$ になる。

関係集合 G'_2 は次の通りとなる。

$$G'_2 = \{R'_1(a, b), R_3(c), R_2\}$$

アルゴリズムのステップ 4 にしたがって、 R'_1 を R_2 に転送して準結合する。その結合結果は低次関係なので、関係集合 G_3 に追加し（ $R_3(c)$ は $R_2(b)$ と同じ理由で G_3 に付加せず）、関係 R_3 と準結合を取る。結果として、次の転送スケジュールを得られる。

$$\text{転送スケジュール： } R_3(a) \longrightarrow R_1 \longrightarrow R_2 \longrightarrow R_3 \longrightarrow$$

そのコストは次の式によって計算できる。

$$\begin{aligned} \text{cost} &= |R_3(a)| + |R_1| \times p_{3a} + |R_2| \times p_{1R'} + |R_3| \times p_{2R'} + 4C_0 \\ &= 30 + 80 \times 2 \times 0.3 + 3000 \times 3 \times 24/10000 + 75000 \times 3 \times 7/1000000 + 5 \times 4 \approx 121 \end{aligned}$$

【例 2】

関係	T	属性 a	属性 b	属性 c	属性 (a, b)	属性 (a, c)	属性 (b, c)	次数	高さ
$R_1(b)$	95(0.95)	-	95(0.95)	-	-	-	-	1	低次
$R_2(a, b)$	70(0.007)	70(0.7)	60(0.6)	-	70(0.007)	-	-	1	低次
$R_3(a, b)$	2000(0.2)	40(0.4)	50(0.5)	-	2000(0.2)	-	-	2	高次
$R_4(a, b, c)$	4800(0.0048)	30(0.3)	60(0.6)	80(0.8)	1500(0.15)	2400(0.24)	4800(0.48)	2	高次

$D_a = D_b = D_c = 100$ とする。各属性の長さは 1 とする。括弧の中は選択度を示す。

表 2. 例題の分散データベース (二)

分散データベースは表 2 の関係から構成されている。 R_1 と R_2 と R_3 と R_4 と属性 (a, b, c) について結合する質問の転送スケジュールをアルゴリズムの処理手順に従って求める。

通信設定のコスト $C_0 = 5$ とする。

アルゴリズムのステップ 2 と 3 によって、次数ごとの関係集合は次の通りである。

$$G'_1 = \{R_4(a), R_3(b), R_4(c), R_1, R_2\}, \quad G_2 = \{R_3, R_4\}$$

関係集合 G'_1 において、アルゴリズムのステップ 4 により、次の転送と関係が選ばれる。

$$R_4(a) \longrightarrow R_2 \tag{18}$$

$R_3(b)$ が削除され、 G'_1 で得られた結果 R'_2 と $R_4(c)$ と R_1 を G_2 に付加すると、関係集合 G'_2 は次の通りとなる。

$$G'_2 = \{R'_2(a, b), R_4(c), R_1, R_3, R_4\}$$

アルゴリズムのステップ 4 にしたがって、次の転送スケジュールが選ばれる。

$$R'_2(a, b) \longrightarrow R_3(a, b) \longrightarrow R_4(a, b, c) \longrightarrow \tag{19}$$

アルゴリズムのステップ 5 では関係 $R'_4(b)$ を R_1 に転送し、準結合をしてから、結果を求めるノードに転送する。

$$R'_4(b) \longrightarrow R_1 \longrightarrow \tag{20}$$

(18), (19), (20) から、最終転送スケジュールは次のようになる。

$$\begin{aligned} \text{転送スケジュール： } R_4(a) \longrightarrow R_2(a, b) \longrightarrow R_3(a, b) \longrightarrow R_4(a, b, c) \longrightarrow \\ R'_4(b) \longrightarrow R_1 \longrightarrow \end{aligned}$$

そのコストは次の式によって計算できる。

$$\text{cost} = |R_4(a)| + |R_2| \times p_{4a} + |R_3| \times p_2 \times p_{4a} + |R_4| \times p_2 \times p_3 + |R_1| \times p'_{4b} + 6C_0 \approx 143$$

5 考察

前章で提案したアルゴリズムを応答時間のアルゴリズムに直すには以下のところで並列化を図ればよい。ステップ3では抽出された関係を適切に全ての関係集合に付加し、ステップ4を並列に実行する方法を取ればよいであろう。

では前章で示したアルゴリズムの効率について論じることとする。まず、すべてのもっとも小さい単一属性を考慮に入れる。単一属性関係はサイズが小さくて、すべての情報を持っている。射影されるもっとも小さい単一属性は、高次の関係から分離される確率が高いため転送コストも低い。次に、2次以上の関係の準結合を考える。この段階では複数属性の準結合また結合が導入される。そのため、関係の属性の間にある対応情報の損失をかなり小さい程度に押さえることができる。かつ、小さい選択度（複数属性を持つ関係の選択度）で他の関係を有効にレデュースすることができる。

すべての関係を単一属性に射影してから結合するという従来の方法（文献[1]～[4]）について考察してみよう。前章の分散データベース（例1）を例に取ることとする。関係 R_2 にすべての単一属性を送って準結合した結果関係はサイズが216であるが、同様にすべての単一属性と関係 R_3 と準結合した結果関係はサイズが11520である。この数値と前章で得られたすべての転送コスト121との差は明らかに非常に大きい。これは単一属性の準結合を用いて高次関係のサイズを効果的に減らされないことを示している。単一属性の限界とも言えよう。

また、関係（複数の属性を持つ）をレデュースに加わることについて論じたい。本論文で提案したアルゴリズムはできるだけ多くの関係をレデュースに入れている。まず、低次の関係を入れて、高次の関係を低次にしてから入れるようになっている。このようにして結合による効果を得ることができた。

各ノードの関係の次数や複数属性の選択度などの情報を管理しなければならないのはネットワークのオーバーヘッドを増やすのではないかという疑問点がある。しかし、実際の分散データベースを考えると各ノードで多くとも数キロバイトの情報量が増えるだけで、大きい問題はないと思われる。その他、高次関係を低次関係にどう分けるか、結果に必要な非結合属性をどう扱うかなどの問題はこれからの研究課題となるだろう。

6 まとめ

本論文でより分散データベースに合うモデルを使って、一般的な分散データベースの質問処理のアルゴリズムを提案した。このアルゴリズムにおける複数属性による準結合をレデュースに入れる方法は非常に有効で、結合演算も取り入れることができた。これらは関係の n 次の次数の概念の導入とその定量化によって実現されている。このような研究はまだ行われていないし、引き続き研究する価値があると思われる。

付録 複数属性を持つ関係の選択度

ここでは複数属性による準結合を行ったときの選択度について考察を行う。関係 $U(a, b, \dots)$ を関係 $R(a, b)$ により準結合を行う。関係 $R(a, b)$ を属性 a, b に分けて別々に結合するではなく、 a と b ペアを送って準結合するものとする。このとき、 $R(a, b)$ のタプルは $D_a \times D_b$ からランダムに t_R 個が選ばれたと考えると、関係 $R(a, b)$ の選択度 p_R は次の式となる。

$$p_R = t_R / (D_a \times D_b) \quad (21)$$

$R(a, b)$ のタプルをこのように選んだとすると、 $R(a, b)$ の a または b の単一属性のタプル数の期待値が求められる[6]。これを使って、特別の場合の近似式を t_a についてのみ与えると、 t_a の期待値は次のようになる。

$$E[t_a] = \begin{cases} D_a & t_R \gg D_a \\ t_R & t_R \ll D_a \end{cases} \quad (22)$$

t_b についても同様である。

これから準結合に用いる関係 $R(a, b)$ に関してそのタプル数 t の他、単一属性のタプル数 t_a, t_b が与えている場合を考えてみよう。このとき t 個のタプルはランダムに選ばれるが、 t_a と t_b が与えているため、前の場合のサブアン

サンプルを考える必要があり、その選択度は(21)と一致しない恐れがある。準結合を受ける関係 $U(a,b)$ のあるタプル x を考える。 x の属性 a の値、 b の値が a 、 b の値の中にある確率 p_a 、 p_b は

$$p_a = t_a/D_a, \quad p_b = t_b/D_b \quad (23)$$

である。この条件が満たされた上で x の中の ab 対の値が $R(a,b)$ の中の ab 対と一致する条件確率 p_{ab} は $R(a,b)$ の ab 対が $t_a \times t_b$ 個の対の中からランダムに選ばれると考えると

$$p_{ab} = t/(t_a \times t_b) \quad (24)$$

となる。(23)の p_a と p_b の積と p_{ab} を掛けて、 $R(a,b)$ の選択度 p は

$$p = p_a \times p_b \times p_{ab} = t/(D_a \times D_b) \quad (25)$$

となり、(21)と一致していることが分かる。このようにして(21)または(25)の選択度の公式は十分な合理性を持つと考えられる。以上2属性のことを考えたが属性数が3以上の場合も同様に拡張できよう。

参考文献

- [1] C. T. Yu, Z. M. Ozsoyoglu and K. Lam. "Optimization in Distributed Query", *J. Computer and System Sciences* 29, 1984, pp.409-445.
- [2] A. R. Hevener and S. B. Yao, "Query processing in Distributed Database System", *IEEE Trans. on Software Engineering*, vol. SE-5, No. 3, May 1979, pp.177-187.
- [3] P. M. G. Apers, A. R. Hevener and S. B. Yao. "Optimization Algorithm for Distributed Queries", *IEEE Trans. on Software Engineering*, vol. SE-9, No. 1, January 1983, pp.57-68.
- [4] P. A. Bernstein, N. Goodman and E. Wong, et al. "Query Processing in a System for Distributed Database(SDD-1)", *ACM Trans. on Database System*, vol. 6, No. 4, December 1981, pp.602-625.
- [5] A. L. P. Chen and V. O. K. Li. "Improvement Algorithm for Semi-join Query Processing Programs in Distributed Database System", *IEEE Trans. on Computers*, vol. C-33, No. 11, November 1984, pp.959-967.
- [6] S. B. Yao. "Approximating Block Accesses in Database Organizations", *Communication of the ACM*, vol. 20, No. 4. April 1977, pp.260-261.
- [7] W. Cellary, Z. Krolikowski and T. Morzy, "Other Comments on 'Optimization Algorithm for Distributed Queries'", *IEEE Trans. on Software Engineering*, vol.14, No. 4, April 1988, pp.439-441.
- [8] 李紅, 佐藤, "分散データベースの質問処理の最適化に関する考察", データベース・システム研究会 64-6, March 1988.
- [9] 李紅, 佐藤, "簡単な分散データベースの質問処理について", 情報処理学会第36回全国大会 3E-10, March 1988.
- [10] 李紅, 佐藤, "分散データベースの木質問の最適化", データベース・システム研究会 64-6, プログラミング言語研究会 26-3, September 1990.
- [11] M. S. Chen and Philip S. Yu, "Determining Beneficial Semijoins for a Join Sequence in Distributed Query Processing", *Seventh international conference on Data Engineering*, April 1991.