

**SIÊU ĐỒ THỊ KẾT NỐI ĐỐI TƯỢNG – MỘT CÁCH TIẾP CẬN  
TRONG TỐI ƯU HOÁ CÂU TRUY VẤN ĐỐI TƯỢNG LỒNG NHAU**

*Lê Mạnh Thạnh*

*Đại học Huế*

*Hoàng Bảo Hùng*

*Sở Thông tin và Truyền thông tỉnh Thừa Thiên Huế*

**TÓM TẮT**

*Trong cơ sở dữ liệu hướng đối tượng, truy vấn đối tượng lồng được sử dụng khá thường xuyên. Các cấu trúc lồng được biểu diễn ở biểu thức điều kiện của truy vấn dưới hai dạng: các truy vấn con lồng hoặc biểu thức đường dẫn có chứa các kết nối ẩn là các tân từ lồng nhau trong mệnh đề where. Đối với truy vấn lồng, khi phân tích ước lượng chi phí của biểu thức đại số lồng nhau, thì việc định giá sẽ cho kết quả chi phí không hiệu quả. Vì vậy, phương pháp của chúng tôi đưa ra trong bài báo này là làm phẳng các truy vấn con trong các truy vấn có cấu trúc lồng nhau sử dụng siêu đồ thị kết nối đối tượng, phương pháp này sẽ làm tăng tính hiệu quả cho việc thực thi xử lý truy vấn trên cơ sở dữ liệu hướng đối tượng.*

**1. Mở đầu**

Để xử lý các tân từ lặp (lồng), Cho W. [3] đưa ra phương pháp ước lượng chi phí phụ thuộc tỷ số giữa số các đối tượng của lớp bắt đầu trong biểu thức đường dẫn và tổng số các đối tượng của lớp, dựa trên mối quan hệ nhiều - nhiều giữa các lớp. Tỷ số này là một trong những tham số lựa chọn trong quá trình thiết kế CSDL vật lý.

Đối với là các truy vấn con lồng, Cluet S. [4] đề xuất phương pháp tối ưu theo hai bước. Đầu tiên, biến đổi các truy vấn ở mức ngôn ngữ nhằm xử lý một cách hiệu quả các biểu thức con chung và các truy vấn con độc lập. Sau đó, các truy vấn được biên dịch thành các biểu thức đại số lồng nhau và áp dụng phương pháp biến đổi đại số. Tuy nhiên, khi phân tích sự ước lượng đối với các vòng lặp lồng nhau trong các biểu thức đại số, chúng ta nhận thấy biểu thức kết quả có chi phí là không hiệu quả. Do đó, phương pháp chúng tôi đề xuất trong phần sau sẽ giải quyết vấn đề xử lý cho các truy vấn con lồng trong giai đoạn “làm phẳng” các truy vấn lồng nhau bằng phương pháp rút gọn siêu đồ thị kết nối đối tượng, giúp cho phép định giá hiệu quả hơn.

Trong bài báo này, xuất phát từ ý tưởng biểu diễn và tối ưu hóa các truy vấn bằng siêu đồ thị của Ullman J.D [9] và Han [5], chúng tôi đưa ra khái niệm siêu đồ thị kết nối đối tượng để biểu diễn các truy vấn đối tượng trong OQL, đặc biệt là xử lý các truy vấn lồng. Từ đó, đề xuất các thuật toán ước lượng các siêu cạnh và thuật toán thu gọn siêu đồ thị kết nối đối tượng.

## 2. Biểu diễn truy vấn đối tượng bằng ký pháp siêu đồ thị

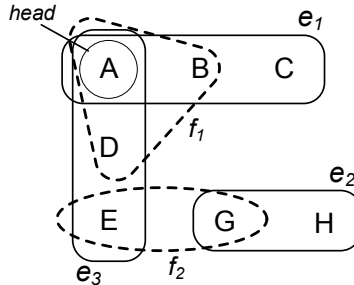
Trước hết, một cách hình thức chúng ta xét định nghĩa của khái niệm siêu đồ thị kết nối đối tượng như sau [6]:

**Định nghĩa.** Siêu đồ thị kết nối đối tượng là một bộ sáu  $\mathcal{C} = (C_H, V_H, E_H, L_H, S_H, lb_H)$ , trong đó:

- (i)  $C_H$  là tập hữu hạn các lớp tham gia trong truy vấn
- (ii)  $V_H$  là tập hữu hạn các nút
- (iii)  $L_H$  là tập hữu hạn các nhãn
- (iv)  $E_H = E_C \cup E_Q$  - tập các siêu cạnh (hữu hạn), trong đó  $E_C, E_Q$  là tập các siêu cạnh biểu diễn các lớp đối tượng và các thành phần của truy vấn.
- (v)  $S_H: V_H \rightarrow E_H$  là ánh xạ khởi tạo các siêu cạnh từ tập các nút
- (vi)  $lb_H: E_H \rightarrow L_H$  là hàm gán nhãn cho siêu cạnh, sao cho  $\forall e \in E_H lb_H(e) \in L_H$

**Ví dụ 2.1.** Xét siêu đồ thị kết nối đối tượng biểu diễn truy vấn sau:

```
select A
from c1, c2, c3
where c1.A = c2.F and (c1.A + c1.B > c3.D) and (c3.E ≤ c2.G)
```



**Hình 1.** Siêu đồ thị kết nối đối tượng của ví dụ 2.1

Trong đó, ta có:

$C_H = \{c_1, c_2, c_3\}$ ,  $c_1 = (A, B, C)$ ,  $c_2 = (G, H)$  và  $c_3 = (D, E, F)$  là các lớp đối tượng.

$V_H = \{A, B, C, D, E, F, G, H\}$ : tập các nút,  $L_H = \{e_1, e_2, f_1, f_2, \text{"head"}\}$ : các nhãn

$E_H = E_C \cup E_Q$ , với  $E_C$  gồm tập các siêu cạnh được gán nhãn  $\{e_1, e_2, e_3\}$  biểu diễn các lớp  $c_1, c_2, c_3$ . Và  $E_Q$  có các siêu cạnh biểu diễn lần lượt là kết quả của truy vấn, biểu thức điều kiện truy vấn tương ứng có nhãn là  $\{f_1, f_2, \text{"head"}\}$ . Đối với điều kiện “ $c_1.A = c_2.F$ ” chúng ta thực hiện “trộn” hai nút đặt nhãn là “A”.

Từ định nghĩa (trong các phần sau, siêu đồ thị kết nối đối tượng được gọi tắt là

siêu đồ thị), chúng ta sử dụng ký pháp siêu đồ thị để biểu diễn cho truy vấn OQL như sau:

- Tập các nút của siêu đồ thị là tập các thuộc tính thuộc các lớp tham gia truy vấn. Mỗi thuộc tính của lớp  $c_i$  được biểu thị bằng một nút. Nếu hai lớp  $c_i$  và  $c_j$  đều có cùng một số các thuộc tính kế thừa từ một siêu lớp nào đó, hoặc chúng cùng kế thừa tất cả các thuộc tính từ một siêu lớp, chúng ta vẫn tạo riêng cho các thuộc tính này các nút khác nhau.

- Các *siêu cạnh* của siêu đồ thị được tạo thành từ các biểu thức điều kiện và các lớp  $c_i$ :

Xét biểu thức điều kiện trong mệnh đề *where*, các biểu thức điều kiện được chia ra các dạng sau:

$$\blacksquare A = a, \quad (1.1)$$

$$\blacksquare A = B, \quad (1.2)$$

$$\blacksquare A \theta B, \theta \in \{<, \leq, \neq, >, \geq\}, \quad (1.3)$$

$$\blacksquare A \theta B, \theta \in \{ \subset, \subseteq, \neq, \supset, \supseteq \} \quad (1.4)$$

trong đó, A, B là thuộc tính của các lớp và  $a$  là hằng.

- Các siêu cạnh là các tập với số lượng nút hữu hạn, biểu diễn lớp, ta gọi là *siêu cạnh đối tượng* và được vẽ bằng một đường khép kín bao quanh các nút của siêu cạnh. Gán nhãn là tên của lớp.
- Đối với mỗi biểu thức điều kiện dạng (1.3) hoặc (1.4), chúng ta sẽ tạo ra một siêu cạnh chứa các thuộc tính có mặt trong biểu thức. Những siêu cạnh này được gọi là *siêu cạnh điều kiện* và chúng được biểu thị bằng đường nét chấm khép kín.
- Điều kiện có dạng (1.1) sẽ trở thành nhãn “ $A = a$ ” của nút biểu diễn thuộc tính tương ứng.
- Biểu thức điều kiện có dạng  $A = B$  (1.2), với A, B là các thuộc tính trong hai lớp (có thể cùng là những thuộc tính được kế thừa từ một siêu lớp nào đó), thì chọn một thuộc tính đại diện và đặt nhãn chung là tên của một trong hai thuộc tính.

- Nếu có hai điều kiện trên cùng một tập thuộc tính, chúng ta phải đặt nhãn riêng cho mỗi siêu cạnh để có thể phân biệt được chúng.

- Các thuộc tính trong mệnh đề *select* được bao trong một đường liền khép kín và gán nhãn là “*head*”, gọi là *siêu cạnh đỉnh*. Siêu cạnh đỉnh tương ứng với một lớp - kết quả của truy vấn.

- *Siêu cạnh kết nhập* chứa các thuộc tính tham gia trong các biểu thức chứa các phép toán {IS, IN, UNION, FORALL, EXIST,...} của các truy vấn con lồng nhau, được vẽ bằng đường nét rời khép kín. Các siêu cạnh kết nhập được gán nhãn tương ứng với tên các phép toán.

Truy vấn đơn chỉ có một khối *select...from...where* (SFW); Truy vấn lồng trong OQL có nhiều hơn một khối SFW. Truy vấn lồng biểu diễn bằng một siêu đồ thị được xây dựng từ các siêu đồ thị của các khối SFW đơn và liên kết với nhau qua các siêu cạnh kết nhập.

Chúng ta biểu diễn hình thức lược đồ đối tượng  $S = (s_1, \dots, s_n)$ ,  $s_i$  là các lớp trong  $S$  và truy vấn đối tượng  $QE = (s_1, \dots, s_m, R, p_1, \dots, p_k)$ , với  $s_i$  ( $i = 1, \dots, m$ ) là các lớp tham gia truy vấn,  $R$  là lớp/kiểu kết quả của truy vấn và  $p_j$  ( $j = 1, \dots, k$ ) là các biểu thức điều kiện ở mệnh đề *where*.

**Thuật toán 2.1:** Khởi tạo siêu đồ thị của truy vấn đối tượng (không chứa truy vấn lồng).

**Vào:** Lược đồ đối tượng  $S = (s_1, \dots, s_n)$

Truy vấn đối tượng  $QE = (s_1, \dots, s_m, R, p_1, \dots, p_k)$

**Ra:** Siêu đồ thị  $\mathcal{A}$

**Phương pháp:**

- (1)  $SC := \emptyset$  //tập chứa các siêu cạnh đối tượng của siêu đồ thị  $\mathcal{A}$
- (2)  $V := (s_1, \dots, s_m)$
- (3) **for**  $s_i \in V$  **do**
- (4)       **if** ( $s_i$  là siêu lớp gốc) **then** //không kế thừa từ các siêu lớp khác
- (5)               Khởi tạo siêu cạnh đối tượng  $e = \mathbf{s}_H(\{s_i\})$  và nhãn  $\mathbf{lb}_H(e)$
- (6)       **else if** ( $s_i$  là lớp kế thừa đơn hoặc kế thừa bội) **then**
- (7)               Xử lý trường hợp xung đột về tên với các thuộc tính kế thừa
- (8)               Khởi tạo lớp  $s_i'$  chứa các thuộc tính của lớp và thuộc tính kế thừa
- (9)               Khởi tạo siêu cạnh đối tượng  $e = \mathbf{s}_H(\{s_i'\})$  và gán nhãn  $\mathbf{lb}_H(e)$
- (10)        $SC := SC \cup e$
- (11)       Khởi tạo siêu cạnh đỉnh  $h = \mathbf{s}_H(\{R\})$  và  $\mathbf{lb}_H(h) = \text{“head”}$
- (12)        $SC := SC \cup h$

- (13)  $SD := \emptyset$  //tập chứa các siêu cạnh điều kiện của siêu đồ thị  $\mathcal{A}$
- (14) **for**  $p_i \in (p_1, \dots, p_k)$  **do**
- (15)           **if** ( $p_i$  có dạng (3.3) và (3.4)) **then**
- (16)                     Khởi tạo siêu cạnh điều kiện  $f = \mathbf{S}_H(\{p_i\})$
- (17)                      $SD := SD \cup f$
- (18)           **else** Gán nhãn cho nút  $\text{lb}_H(e) = "= a"$
- (19)  $\mathcal{A} := SC \cup SD$

Từ định nghĩa, chúng ta khẳng định thuật toán 2.1 là đúng đắn và độ phức tạp tính toán của thuật toán là  $O(n^2)$ , với  $n$  là số lớp tham gia trong truy vấn.

Trên cơ sở của thuật toán 2.1, chúng ta xây dựng thuật toán khởi tạo siêu đồ thị biểu diễn các truy vấn lồng. Với các bước xác lập siêu đồ thị đơn, chúng ta tạo siêu đồ thị kết quả từ các liên kết của các siêu đồ thị đơn với các siêu cạnh kết nhập.

**Thuật toán 2.2:** Khởi tạo siêu đồ thị của truy vấn lồng OQL.

**Vào:** Lược đồ đối tượng  $S = (s_1, \dots, s_n)$

Truy vấn  $QE = (s_1, \dots, s_m, R, p_1, \dots, p_k)$ , tập  $TT \subseteq \{\text{is, in, union, diff, forall, exists}\}$  là tập các toán tử tham gia trong mệnh đề *where* của truy vấn  $QE$ .

**Ra:** Siêu đồ thị  $\mathcal{A}$ .

**Phương pháp:**

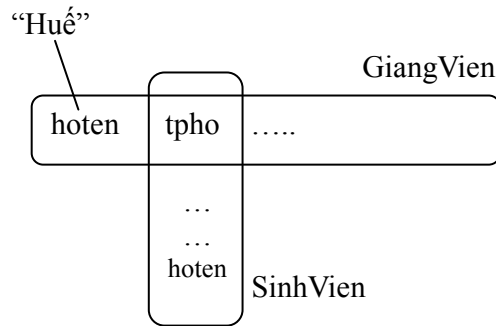
- (1)  $\mathcal{A} := \emptyset$
- (2) **for** (mỗi truy vấn con  $QE_i$  trong truy vấn  $QE$ ) **do**
- (3)           Khởi tạo siêu đồ thị  $\mathcal{A}_i$  với  $QE_i$  (Thuật toán 2.1)
- (4)            $\mathcal{A} := \mathcal{A} \cup \mathcal{A}_i$
- (5) **for** (mỗi toán tử  $t_i \in TT$ ) **do**
- (6)           Khởi tạo siêu cạnh kết nhập  $g$  có nhãn là  $t_i$  chứa siêu cạnh đỉnh của siêu đồ thị ở vế phải của  $t_i$  và thuộc tính ở vế trái của  $t_i$
- (7)            $\mathcal{A} := \mathcal{A} \cup g$

**Ví dụ 2.2.** Tìm tên của tất cả sinh viên ở cùng thành phố với giảng viên có tên là “Huế”.

```
define SinhVien as p1
select (p1.name)
from p1, p2 GiảngVien
```

where p1.tpho = p2.tpho AND p2.hoten = “Huế”

Truy vấn trong ví dụ 2.2 chứa các kết nối dựa trên giá trị (p1.tpho = p2.tpho) và được gán nhãn là “tpho”, trong 2 nút có nhãn là “tpho”.

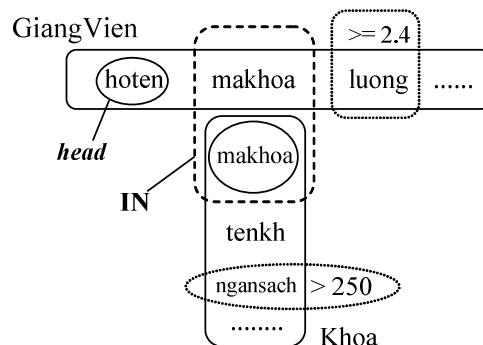


**Hình 2.2.** Siêu đồ thị biểu diễn cho ví dụ 2.2

**Ví dụ 2.3.** Xét truy vấn cho biết tên các cán bộ giảng viên (*cbgv*) ở khoa có ngân sách được cấp lớn hơn 250 (đơn vị tính: triệu đồng) và có mức lương lớn hơn hoặc bằng 2.4.

```
select e.hoten
from      GiangVien as e
where e.luong >= 2.4
AND e.makhoa IN ( select s.makhoa
                  from   Khoa as s
                  where  s.ngansach > 250)
```

Siêu đồ thị của ví dụ 2.3 được khởi tạo như sau: Các siêu cạnh đối tượng biểu diễn các lớp GiangVien và Khoa. Đối với các siêu cạnh đỉnh, chúng ta có hai siêu cạnh đỉnh: *e.hoten* - siêu cạnh đỉnh (kết quả của truy vấn), *s.makhoa* - siêu cạnh đỉnh của khối SFW lồng. Hai siêu cạnh điều kiện *e.luong* >= 2.4, *s.ngansach* > 250 và siêu cạnh kết nhập *e.makhoa* được gán nhãn là **IN**.



**Hình 2.2.** Biểu diễn siêu đồ thị của ví dụ 2.3

### 3. Phương pháp ước lượng siêu đồ thị

#### 3.1. Ước lượng các siêu cạnh [6]

Biểu diễn hình thức cho siêu đồ thị của truy vấn hướng đối tượng là dãy các sự kiện:  $\mathcal{C} = (E_1, E_2, \dots, E_n)$ , trong đó, các sự kiện  $E_i$  có thể là siêu cạnh đối tượng, siêu cạnh điều kiện hoặc siêu cạnh kết nhập.

Lớp dẫn xuất thu được sau tác động của một sự kiện  $E_j$  được ký hiệu là  $Lớp\_dẫn\_xuất(E_1, \dots, E_j)$ , trong đó  $E_1$  phải là siêu cạnh đối tượng (trường hợp siêu đồ thị chỉ có một siêu cạnh thì nó phải là siêu cạnh đối tượng).

Thủ tục  $Ước\_lượng\_siêu\_cạnh$  nhận vào lớp dẫn xuất thu được sau tác động của sự kiện  $E_{j-1}$  và sự kiện  $E_j$ , kết quả của thủ tục là lớp dẫn xuất với sự kiện  $E_j$  trong  $\mathcal{C}$ .

**Procedure**  $Ước\_lượng\_siêu\_cạnh(Lớp\_dẫn\_xuất(E_1, \dots, E_{j-1}), E_j)$

**Vào:**  $Lớp\_dẫn\_xuất(E_1, \dots, E_{j-1})$  và sự kiện  $E_j$

**Ra:**  $Lớp\_dẫn\_xuất(E_1, \dots, E_j)$

**Phương pháp:**

(1) Khởi tạo,  $Ước\_lượng\_siêu\_cạnh(E_1)$  cho kết quả:  $Lớp\_dẫn\_xuất(E_1) = c_1$ , trong đó,  $c_1$  là lớp tương ứng với siêu cạnh đối tượng  $E_1$ .

(2) **if** ( $E_j$  là một điều kiện hoặc siêu cạnh điều kiện) **then**

$Lớp\_dẫn\_xuất(E_1, \dots, E_j) = \sigma_F(Lớp\_dẫn\_xuất(E_1, \dots, E_{j-1}))$

với  $F$  là biểu thức điều kiện tương ứng với  $E_j$

(3) **if** ( $E_j$  là siêu cạnh đối tượng của lớp  $C_j$  có giao với siêu đồ thị) **then**

$Lớp\_dẫn\_xuất(E_1, \dots, E_j) = Lớp\_dẫn\_xuất(E_1, \dots, E_{j-1}) \bowtie C_j$

(4) **if** ( $E_j$  là siêu cạnh đối tượng không giao với siêu đồ thị) **then**

$Lớp\_dẫn\_xuất(E_1, \dots, E_j) = Lớp\_dẫn\_xuất(E_1, \dots, E_{j-1}) \times C_j$

Trong ví dụ 2.2,  $S = (GiangVien, SinhVien)$ , ta có:

$Lớp\_dẫn\_xuất(GiangVien, SinhVien) = Lớp\_dẫn\_xuất(GiangVien) \bowtie SinhVien$ .

Khi tất cả các siêu cạnh trong siêu đồ thị  $\mathcal{C}$  được ước lượng bằng cách tác động lần lượt các sự kiện để thu được các lớp dẫn xuất. Lớp dẫn xuất kết quả sẽ được chiếu lên tập các thuộc tính trong siêu cạnh đỉnh, đây chính là câu trả lời của truy vấn.

**Thuật toán 3.1: Ước lượng các siêu cạnh của siêu đồ thị.**

**Vào:** Siêu đồ thị  $\mathcal{C} = (E_1, E_2, \dots, E_n)$ ,  $R$  là siêu cạnh đỉnh.

**Ra:** Lớp kết quả của truy vấn.

**Phương pháp:**

- (1) Biểu diễn siêu đồ thị  $\mathcal{C} = (E_1, E_2, \dots, E_n)$ , với dãy các sự kiện  $E_i$
- (2) **for**  $j = 1$  **to**  $n$  **do**
- (3)       **Call** *Ước lượng siêu cạnh*(*Lớp dẫn xuất*( $E_1, \dots, E_{j-1}$ ),  $E_j$ )
- (4)       **Bổ sung** *Lớp dẫn xuất*( $E_1, \dots, E_j$ ) vào  $\mathcal{C}$
- (5)  $\mathcal{C} = \pi_R(\text{Lớp dẫn xuất}(E_1, \dots, E_n))$

**Mệnh đề.** Thuật toán 3.1 dừng sau hữu hạn bước và cho câu trả lời đúng.

*Chứng minh.*

Rõ ràng số các sự kiện trong  $\mathcal{C}$  là hữu hạn, do đó thuật toán 3.1 sẽ dừng sau  $n$  lần tác động các sự kiện  $E_j$  lên các lớp dẫn xuất,  $n$  là số sự kiện trong siêu đồ thị  $\mathcal{C}$ .

Để chứng minh rằng thuật toán 3.1 trả về câu trả lời đúng của truy vấn đã cho, chúng ta chứng minh quy nạp như sau:

Trường hợp cơ sở:  $n = 1$ , thì  $\mathcal{C} = (E_1)$ ,  $E_1$  là siêu cạnh đối tượng, ta có:

$$\mathcal{C} = \pi_R(E_1) = \pi_R(C_1) - \text{là câu trả lời của truy vấn.}$$

Giả sử lớp dẫn xuất thứ  $k$  thu được sau tác động của sự kiện  $E_k$  là ước lượng của  $k$  siêu cạnh trong siêu đồ thị (*Lớp dẫn xuất*( $E_1, \dots, E_k$ )). Mặt khác, lớp dẫn xuất thu được ở bước thứ  $k$  lại là đầu vào cho bước ước lượng thứ  $(k+1)$ , do đó nếu  $k = n$  thì sau  $n$  bước ước lượng siêu cạnh, ta có lớp dẫn xuất thu được là:

$$\text{Ước lượng siêu cạnh}(\text{Lớp dẫn xuất}(E_1, \dots, E_{n-1}), E_n) = \text{Lớp dẫn xuất}(E_1, \dots, E_n)$$

Sau đó chiếu lên siêu cạnh đỉnh ta có kết quả của truy vấn:

$$\mathcal{C} = \pi_R(\text{Lớp dẫn xuất}(E_1, \dots, E_n)) - \text{kết quả của truy vấn. } \square$$

Ví dụ, xét truy vấn trong ví dụ 2.2, chúng ta có dãy sự kiện trong siêu đồ thị  $\mathcal{C} = (\text{GiangVien}, \text{hoten} = \text{“Huế”}, \text{SinhVien})$ .

Áp dụng thuật toán 3.1, ta có các bước ước lượng như sau: (1) Khởi tạo lớp dẫn xuất tương ứng với siêu cạnh đối tượng *GiangVien*, (2) Áp dụng điều kiện chọn “hoten = “Huế” trên siêu cạnh đối tượng *GiangVien*, (3) Ước lượng siêu cạnh đối tượng *SinhVien*, là kết nối truyền thống dựa trên giá trị, (4) Chiếu lớp dẫn xuất thu được sau phép kết nối lên siêu cạnh đỉnh (hoten).

Chúng ta có thể nhận thấy từ ví dụ 2.4, trật tự sắp xếp các siêu cạnh trong  $\mathcal{C}$  sẽ cho dãy các bước ước lượng khác nhau, tương ứng với sự sắp thứ tự các phép toán được thực thi. Đây sẽ là một trong các tham số xác định không gian tìm kiếm các phương án thực thi truy vấn.



Trong thuật toán 3.1, chúng ta chưa xử lý cho trường hợp các siêu cạnh kết nhập, tức là, xét các siêu đồ thị biểu diễn các truy vấn đối tượng lồng. Các truy vấn lồng có thứ tự thực hiện các truy vấn con theo trật tự từ “trong ra ngoài”, nghĩa là, các truy vấn ở cấp sâu nhất sẽ được thực hiện trước. Giả sử siêu đồ thị biểu diễn các truy vấn đối tượng lồng được mô tả một cách hình thức là một dãy các sự kiện chúng ta xây dựng thuật toán thu gọn siêu đồ thị  $\mathcal{C} = (E_1, E_2, \dots, E_i, EA_j, E_{i+1}, \dots, E_k, \dots)$ , trong đó  $E_i$  là các siêu cạnh đối tượng hoặc siêu cạnh điều kiện,  $EA_j$  là các siêu cạnh kết nhập.

Chúng ta mở rộng thuật toán 3.1 bằng việc xử lý các siêu cạnh kết nhập như sau: Trước hết, chúng ta ước lượng tất cả các siêu cạnh  $E_l$  ( $l = i + 1, \dots, k$ ) sau siêu cạnh kết nhập  $EA_j$ , và không mất tính tổng quát chúng ta giả sử  $E_k$  là siêu cạnh cuối cùng được ước lượng (trước khi ước lượng siêu cạnh kết hợp  $EA_j$ ). Sau đó, tác động sự kiện  $EA_k$  đối với lớp dẫn xuất thu được sau khi tác động sự kiện thứ  $E_k$  ( $EA_j$ ). Như vậy,  $\mathcal{C}$  được viết lại là  $\mathcal{C} = (E_1, E_2, \dots, E_i, EA_j, E_{i+1}, \dots, E_k, EA_j, \dots)$ ,  $EA_j$  là lớp dẫn xuất kết quả sau khi ước lượng siêu cạnh kết nhập.

### Thuật toán 3.2: Thu gọn siêu đồ thị

**Vào:** Siêu đồ thị  $\mathcal{C} = (E_1, E_2, \dots, E_i, EA_j, E_{i+1}, \dots, E_k, \dots)$ ,  $R$  là siêu cạnh đỉnh.

**Ra:** Lớp kết quả của truy vấn.

**Phương pháp:**

- (1) Biểu diễn siêu đồ thị  $\mathcal{C} = (E_1, E_2, \dots, E_i, EA_j, E_{i+1}, \dots, E_k, \dots)$
- (2)  $i := 1$
- (3) **repeat**
- (4)     **if** ( $EA_j$  là siêu cạnh kết nhập) **then**
- (5)         **for**  $l = i + 1$  **to**  $k - 1$  **do**
- (6)             **Call** *Ước lượng siêu cạnh*(Lớp\_dẫn\_xuất( $E_j$ ),  $E_{j+1}$ )
- (7)             **Bổ sung** *Lớp dẫn xuất*( $E_l, \dots, E_k$ ) vào  $\mathcal{C}$  trước  $EA_j$
- (8)     **else** **Call** *Ước lượng siêu cạnh*(Lớp\_dẫn\_xuất( $E_1, \dots, E_{i-1}$ ),  $E_i$ )
- (9)     **Bổ sung** *Lớp dẫn xuất*( $E_1, \dots, E_i$ ) vào  $\mathcal{C}$
- (10)     **inc**( $i$ )
- (11) **until** (ước lượng tất cả các siêu cạnh trong  $\mathcal{C}$ )
- (12)  $\mathcal{C} = \pi_R(\text{Lớp\_dẫn\_xuất}(E_1, \dots, E_k, \dots))$

Từ mệnh đề trên, chúng ta suy ra thuật toán 3.2 là dừng và cho kết quả là câu trả lời đúng. Độ phức tạp tính toán của thuật toán 3.2 có thời gian đa thức.

### 3.2. Không gian tìm kiếm của truy vấn

Các phương án thực thi truy vấn trong không gian tìm kiếm được xét trên các khả năng lựa chọn, ước lượng của các lớp và các siêu cạnh thuộc siêu đồ thị. Từ đó, với các thuật toán ước lượng và thu gọn siêu đồ thị (thuật toán 3.1 và 3.2), chúng ta sẽ xây dựng không gian tìm kiếm các phương án thực thi truy vấn như sau (thuật toán 3.3).

Chúng ta dùng tập các danh sách với các phần tử chứa các thành phần của siêu đồ thị, sau đó lần lượt duyệt tập các danh sách tương ứng để xác định các phương án.

**Thuật toán 3.3.** Không gian tìm kiếm của truy vấn.

**Vào:** Siêu đồ thị  $\mathcal{C}$

**Ra:** Không gian tìm kiếm với tổng số các phương án thực thi truy vấn

**Phương pháp:**

(1) Sắp xếp các lớp, các siêu cạnh đối tượng, điều kiện và các siêu cạnh kết nhập vào tập danh sách  $\{L_l\}$

// Bước 1: Ước lượng các siêu cạnh điều kiện và siêu cạnh kết nhập

(2) **for** mỗi danh sách  $L_l$  **do**

(3)           **for** mỗi siêu cạnh  $E$  **do**

(4)                   **if**  $E$  là siêu cạnh kết nhập **then**

(5)                                   Bổ sung  $EA_j$  vào  $L_l$  sau siêu cạnh cuối cùng  $E_k$

//Thuật toán 3.2

(6)                   **else if**  $E$  là siêu cạnh điều kiện **then**

(7)                                   Ước lượng các siêu cạnh điều kiện //Thuật toán 3.1

(8)           Kết quả thu được là danh sách  $\{L_l'\}$

// Bước 2: Ước lượng các siêu cạnh đối tượng

(9) **for** mỗi danh sách  $L_l'$  **do**

(10)           **for** mỗi siêu cạnh đối tượng **do**

(11)                   Thực hiện các ước lượng với các siêu cạnh điều kiện tương ứng

(12)           Kết quả lưu ở danh sách  $\{L_2\}$

// Bước 3: Ước lượng các kết nối

(13) **for** mỗi danh sách  $L_2$  **do**

- (14) **for** mỗi siêu cạnh **do**
- (15) Ước lượng các kết nối trên các lớp
- (16) Kết quả là danh sách  $\{L_3\}$ ;

Tập các danh sách  $\{L_3\}$  là không gian tìm kiếm của các phương án thực hiện truy vấn.

Chúng ta ký hiệu **KGTK** là tổng số các phương án thực thi truy vấn đối tượng trong không gian tìm kiếm.

**Định lý.** Không gian tìm kiếm trong thuật toán 3.3 có tổng số các phương án thực thi truy vấn là:

$$(p + q! + 2^{q-1}) \leq \mathbf{KGTK} \leq (q! + p2^{q-1})$$

trong đó,  $p$  là số các siêu cạnh của siêu đồ thị,  $q$  là chi phí của các phép toán đại số.

*Chứng minh.*

Không gian tìm kiếm trong thuật toán 3.3 được xác định qua các tham số sau: (i) Thứ tự sắp xếp các siêu cạnh trong danh sách, (ii) Chi phí thực hiện các phép toán và (iii) Chi phí các kết nối trong truy vấn. Đối với (i), nếu  $p$  là số các siêu cạnh thì số các giao hoán về thứ tự của các siêu cạnh trong danh sách  $L_l$  là  $p!$ . Tham số trong (ii) phụ thuộc vào thiết kế của hệ thống và chi phí của các kết nối trên  $p$  siêu cạnh là  $2^{p-1}$  (iii). Vậy tổng số các phương án thực thi truy vấn là:

$$(p + q! + 2^{q-1}) \leq \mathbf{KGTK} \leq (q! + p2^{q-1})$$

Trong trường hợp xấu nhất, các phương án trong các bước ước lượng là độc lập nhau và danh sách kết quả của bước này là đầu vào của bước sau thì  $\mathbf{KGTK} = q! + p2^{q-1}$ , còn không  $\mathbf{KGTK} = q! + p + 2^{q-1}$ .  $\square$

#### 4. Kết luận

Tối ưu hóa truy vấn đối tượng đã được các tác giả quan tâm nghiên cứu với nhiều kết quả được xây dựng trên các phương án tiếp cận khác nhau như phương pháp tối ưu hóa dựa trên ước lượng chi phí xử lý truy vấn, tối ưu hóa biểu thức đường dẫn trong truy vấn đối tượng, các phương pháp phân tách ngang/dọc các lớp đối tượng... Trong bài báo này, chúng tôi đề xuất phương pháp tối ưu hóa truy vấn đối tượng được phát triển từ cách tiếp cận siêu đồ thị được Ullman J.D [9] và Han [5] đề xuất để giải quyết cho lớp các truy vấn đối tượng lồng với thuật toán ước lượng trên siêu cạnh đối tượng có chi phí xử lý truy vấn hiệu quả hơn so với phương pháp biến đổi biểu thức đại số đối tượng và phương pháp tối ưu biểu thức đường dẫn. Mặt khác, việc sử dụng ký pháp siêu đồ thị đối tượng có thể biểu diễn và tối ưu cho một lớp các truy vấn trên các đối tượng phức. Vấn đề xây dựng thuật toán tổng quát để rút gọn siêu đồ thị đối

tượng và sự kết hợp giữa không gian tìm kiếm với mô hình ước lượng chi phí truy vấn cho phép xác định được phương án thực thi truy vấn tối ưu sẽ được chúng tôi nghiên cứu và giới thiệu trong các bài báo sau.

## TÀI LIỆU THAM KHẢO

1. Đoàn Văn Ban, Lê Mạnh Thanh và Hoàng Bảo Hùng. *Sự tương đương trong biểu diễn giữa ngôn ngữ truy vấn OQL và đại số đối tượng*, Tạp chí Tin học và Điều khiển học, 20(3), (2004), 257–269.
2. Cattel R.G.G., Barry D.K.. *The Object Database Standard: ODMG 3.0*, Morgan Kaufmann Publishers, 2000.
3. Cho Wan-Sup, Han Wook-Shin, Hong Ki-Hyung and Whang Kyu-Young. *Estimating Nested Selectivity in Object-Oriented Databases*, ACM, (2000), 94–101.
4. Cluet, Sophie and Moerkotte, Guido. *Nested Queries In Object Bases*, In Fifth International Workshop on Database Programming Languages, Italy, 1995.
5. Han, Jia Liang. *Optimizing Relational Queries in Connection Hypergraphs: Nested Queries, Views, and Binding Propagations*. The VLDB Journal, 7, (1998), 1–11.
6. Lê Mạnh Thanh, Đoàn Văn Ban, Hoàng Bảo Hùng. *Phương pháp ước lượng các truy vấn lồng trong cơ sở dữ liệu hướng đối tượng bằng siêu đồ thị kết nối*. Chuyên san Tạp chí Bưu chính Viễn thông và Công nghệ thông tin, “Các công trình nghiên cứu - Triển khai Viễn thông và Công nghệ thông tin”, 14, (2005), 43–49.
7. Trigoni A. and Bierman G.M.. *Inferring the Principal Type and the Schema Requirements of an OQL Query*. In 18th British National Conference on Databases (BNCOD), (2001), 185–201.
8. Trigoni A.. *Semantic Optimization of OQL Queries*, Technical Report, Number 547, University of Cambridge, Computer Laboratory, UCAM-CL-TR-547, ISSN 1476-2986, 2002.
9. Ullman J. D. : *Principles of Database and Knowledge base Systems*. Vol. I: Classical Database Systems, Computer Science Press. New York, 1988. Vol. II: The New Technologies, Computer Science Press. New York, 1989.

# OBJECT CONNECTION HYPERGRAPHS – AN APPROACH FOR NESTED OBJECT QUERY OPTIMIZATION

*Le Manh Thanh*

*Hue University*

*Hoang Bao Hung*

*Department of Information and Communications of Thua Thien Hue Province*

## SUMMARY

*The nested object queries are regularly used in Object-Oriented Databases (OODB). Nested structures are put in conditional expressions of the queries in two forms: nested sub-queries or path expression containing hidden joins – nested predicates in where clause. For nested queries, when analyzing the estimated cost of the nested algebraic expression, the expression evaluation result gives out an ineffective cost. Therefore, our method proposed in this paper will solve the problems by leveling nested sub-queries in the nested queries. This method will increase the effectiveness of the query processing cost – We use object connection hypergraphs to present nested queries.*