*Tạp chí Tin học và Điều khiển học, T.25, S.2 (2009), 178–187*

# GOM CỤM CÁC ĐỐI TƯỢNG TRONG CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG SỬ DỤNG MA TRẬN KHOẢNG CÁCH

ĐOÀN VĂN BAN[1], TRƯƠNG NGỌC CHÂU[2]

[1] *Viện Công nghệ thông tin, Viện Khoa học và Công nghệ Việt Nam*
[2] *Trường Đại học Bách khoa, Đại học Đà Nẵng*

**Abstract.** In this article, we focus on researching the types of the objects relation in the object oriented database before we cluster them in the secondary storage. Based on the distance matrix of the objects joined in the relations, we propose an algorithm to order the objects into a sequence with the shortest sum of the distance of these objects. After this sequence is built, we map it into the continuous blocks in the secondary storage.

**Tóm tắt.** Bài báo tập trung nghiên cứu các loại quan hệ giữa các đối tượng trong cơ sở dữ liệu hướng đối tượng trước khi gom cụm chúng để lưu trữ vào bộ nhớ thứ cấp. Dựa trên ma trận khoảng cách giữa các đối tượng tham gia trong các quan hệ, tác giả đã đề xuất thuật toán sắp xếp thứ tự các đối tượng vào dãy sao cho tổng khoảng cách giữa chúng là cực tiểu trước khi lưu trữ chúng vào bộ nhớ thứ cấp.

## 1. MỞ ĐẦU

Gom cụm dữ liệu trong lưu trữ là một vấn đề quan trọng trong thiết kế cơ sở dữ liệu vật lý nói chung. Cách thức lưu trữ các dữ liệu có liên quan đến đơn vị lưu trữ thứ cấp nhằm nâng cao hiệu suất cho các truy vấn dữ liệu là rất cần thiết. Do đó, gom cụm có thể cải tiến hiệu suất của các hệ thống cơ sở dữ liệu và đây được xem là vấn đề cần giải quyết chính trong các hệ quản trị cơ sở dữ liệu. Trong các hệ thống cơ sở dữ liệu hướng đối tượng, việc gom cụm là phức tạp vì tồn tại nhiều mối quan hệ giữa chúng. Một đối tượng có thể vừa là thể hiện của một lớp vừa là thành phần của đối tượng phức hợp. Bên cạnh các quan hệ kết nhập (aggregation) và khái quát hóa (generalization), các quan hệ cấu trúc khác như phiên bản (version) và cấu hình (configuration) cũng có thể song song tồn tại. Trong trường hợp này, các đối tượng có nhiều mối quan hệ chặt chẽ với nhau. Điều này ảnh hưởng đến các phương thức gom cụm các đối tượng. Để giải quyết vấn đề này, chúng tôi đề xuất một giải pháp nhằm dàn xếp các quan hệ có thể giữa các đối tượng và sau đó sắp xếp các đối tượng này vào một dãy gom cụm. Sau khi dãy được xây dựng, chúng tôi ánh xạ dãy các đối tượng được gom cụm vào đơn vị lưu trữ thứ cấp mà vẫn bảo toàn các quan hệ giữa các đối tượng.

Bài báo được tổ chức như sau: Mục 2 trình bày một số kỹ thuật gom cụm đối tượng trong cơ sở dữ liệu hướng đối tượng và đề ra giải pháp tiếp cận khác bằng ma trận khoảng cách; Mục 3 đưa ra khái niệm và cách xây dựng ma trận khoảng cách dựa trên các mối quan

hệ giữa các đối tượng; Mục 4 đề xuất một thuật toán gom cụm các đối tượng dựa trên ma trận khoảng cách, và cuối cùng là kết luận.

## 2. MỘT SỐ GIẢI PHÁP GOM CỤM CÁC ĐỐI TƯỢNG

Để thực thi truy vấn dữ liệu trong hệ thống một cách hiệu quả, thông thường phụ thuộc vào nhiều yếu tố, như: tốc độ của CPU, bộ nhớ, chi phí vào/ra (I/O) đĩa và các tài nguyên truyền thông. Để tối ưu hóa truy vấn, thông thường người ta chú trọng nhiều vào việc tối ưu hóa ngữ nghĩa của câu truy vấn [8] và ít khi đề cập đến việc tối ưu hóa các chi phí I/O và các yếu tố khác. Khi thực thi một truy vấn thì trước tiên dữ liệu được truyền từ đĩa vào bộ nhớ chính theo đơn vị khối (block) và việc truy xuất dữ liệu trên đĩa thường chậm hơn nhiều so với truy xuất dữ liệu trong bộ nhớ chính. Do đó, cần thiết một chiến lược nhằm nâng cao truy xuất khối đĩa đối với các cơ sở dữ liệu lớn không thể lưu trữ ở bộ nhớ chính. Một trong những chiến lược hiệu quả là lưu trữ các thông tin có liên quan với nhau vào các khối kề nhau, chiến lược này còn được gọi là gom cụm dữ liệu. Thuật toán gom cụm được sử dụng phổ biến là thuật toán K-Means [3], thuật toán này thực hiện phân cụm dựa vào các yếu tố do người dùng cung cấp như số cụm, $k$ điểm được chọn làm trọng tâm. Thuật toán này không phù hợp cho việc gom cụm các đối tượng trong môi trường tồn tại nhiều mối quan hệ. Một tiếp cận khác được đề xuất trong [9], các tác giả đưa ra các kỹ thuật gom cụm dữ liệu nhằm cực tiểu thời gian truy cập, cách tiếp cận gom cụm đối tượng dựa vào các tổ hợp xác suất và có thể gom cụm các đối tượng trong các liên kết bội. Ngoài ra, các tác giả trong [1, 2, 5, 7] cũng đã đưa ra nhiều giải pháp để lựa chọn chiến lược gom cụm các đối tượng:

1. Gom cụm tất cả các đối tượng thuộc cùng một lớp vào trong cùng phân đoạn (segment) của các trang đĩa theo quan hệ thành viên instance-of.
2. Gom cụm tất cả các đối tượng thuộc cùng một phân cấp lớp theo quan hệ is-a.
3. Gom cụm tất cả đối tượng có cùng tham chiếu bên trong theo quan hệ bộ phận part-of. Trong trường hợp này các đối tượng có thể thuộc về các lớp khác nhau.
4. Kết hợp kỹ thuật gom cụm của 2 và 3.

Mỗi cách gom cụm có thể là tối ưu cho loại truy vấn này nhưng không tối ưu cho loại truy vấn khác. Ví dụ, gom cụm tất cả các đối tượng theo quan hệ part-of vào cùng 1 khối (block) sẽ gây ra trường hợp các đối tượng của cùng một lớp sẽ được lưu trữ trải ra trên nhiều trang thay vì lưu trữ chúng trong cùng một khối. Tuy nhiên, tính chất quan hệ bội giữa các đối tượng có nhiều mối quan hệ là một ràng buộc vốn có trong các hệ thống hướng đối tượng. Do đó, cần có một phương pháp xử lý hiệu quả cho việc thực hiện gom cụm các đối tượng có các quan hệ bội, nhằm nâng cao hiệu suất cho các truy vấn của người dùng là cần thiết.

Steven Yi-Cheng Tu và Daniel J. Bueher [6] đã đề xuất cách sử dụng khoảng cách để đo mức độ quan hệ của các đối tượng. Mỗi đối tượng theo phân loại của các quan hệ mà nó có

thể tham gia, có thể được trình bày bởi $n$-bộ như sau: (category 1,, category n), nếu có n quan hệ trong hệ thống.

**Ví dụ 1**. Cho cơ sở dữ liệu đối tượng được minh họa như sau:

| | |
|---|---|
| define class GIAOVIEN: | define class KHOA: |
| type tuple( hoTen: String; | type tuple( tenKhoa: String; |
| hocVi: String; | truongKhoa: GIAOVIEN; |
| khoa: KHOA; | giaoVien: set(GIAOVIEN); |
| ) | ) |
| end GIAOVIEN | end KHOA |

Giả sử chúng ta có các đối tượng được thể hiện trong cơ sở dữ liệu trên như sau:

$O_1$ = ($i_1$, atom, Trần Văn Nam)

$O_2$ = ($i_2$, atom ,Tiến sĩ)

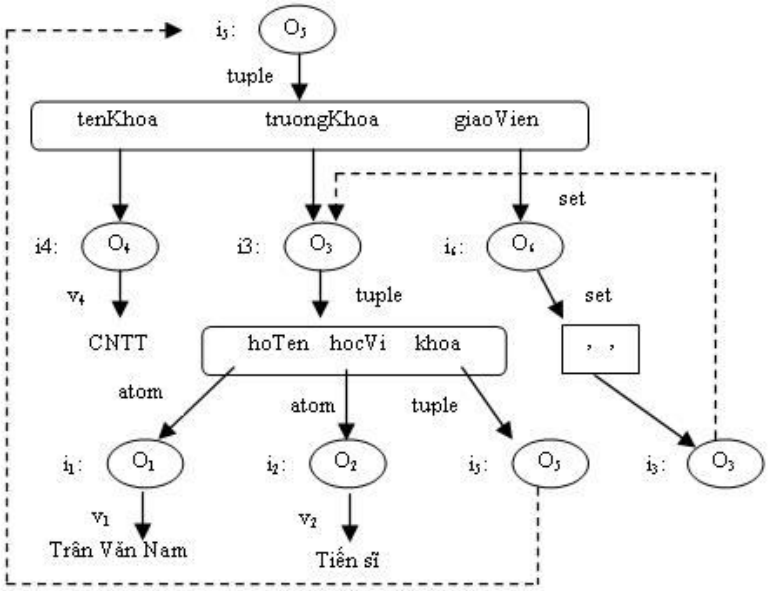
$O_3$ = ($i_3$, tuple, <hoTen: $i_1$, hocVi: $i_2$, khoa: $i_5$ >)

$O_4$ = ($i_4$, atom, CNTT)

$O_5$ = ($i_5$, tuple, <tenKhoa: $i_4$, truongKhoa: $i_3$, giaoVien: $i_6$ >)

$O_6$ = ($i_6$, set, $i_3$)

và tập các đối tượng phức GIAOVIEN kí hiệu là $CO_{gv}$, KHOA kí hiệu là $CO_{khoa}$, lớp chứa các thể hiện của đối tượng có giá trị nguyên thủy kí hiệu là C. Khi đó, đối tượng $O_3$ là bộ (GIAOVIEN, $CO_{khoa}$), có thể được hiểu đối tượng này là một thể hiện của lớp GIAOVIEN, và thuộc đối tượng phức $CO_{khoa}$. Sơ đồ mô tả đối tượng $O_5$ được mô tả như trong hình 1.




*Hình 1.* Biểu diễn đối tượng $O_5$ dưới dạng đồ thị

Vấn đề là hầu hết các giá trị trong mỗi category trong $n$-bộ là những dữ liệu định tính và rời rạc. Vì vậy, để tính khoảng cách giữa các cặp đối tượng ta cần phải có phương thức để chuyển đổi $n$-bộ vào dữ liệu kiểu số tương ứng, để từ đó ta có thể đo được khoảng cách giữa chúng. Có hai ràng buộc cơ bản phải được thỏa mãn khi chuyển đổi $n$-bộ vào dữ liệu

kiểu số:

1. Bất kỳ một cấu tính toán nào cũng phải bảo đảm rằng các đối tượng có liên quan chặt chẽ với nhau hơn thì có khoảng cách giữa chúng gần nhau hơn.

2. Nếu hai đối tượng là không thể so sánh được mối liên quan bên trong thì không nên có sự so sánh bất kỳ nào đối với khoảng cách sau khi biến đổi.

Có thể hình thức hóa các ràng buộc trên qua các ký hiệu như sau:

$\Theta$: quan hệ ngữ nghĩa giữa hai đối tượng, thể hiện mối quan hệ giữa các đối tượng.

$\alpha$: toán tử liên quan chặt chẽ hơn hay thân thiện hơn.

Ràng buộc 1: nếu $\Theta(O_1, O_2)\alpha\Theta(O_1, O_3)$ thì $d(O_1, O_2) < d(O_1, O_3)$.

Ràng buộc 2: nếu $not(\Theta(O_1, O_2)\alpha\Theta(O_1, O_3))$ thì $not(d(O_1, O_2) < d(O_1, O_3))$.

**Ví dụ 2.** Xét cơ sở dữ liệu trong Hình 1, ta có hai đối tượng $O_1$, $O_2$ cùng là thể hiện của lớp C và cùng thuộc $CO_{gv}$, nhưng $O_1$ và $O_4$ chỉ cùng thể hiện của C, nên:

$\Theta(O_1, O_2)\alpha\Theta(O_1, O_4)$ suy ra $d(O_1, O_2) < d(O_1, O_4)$ ($d$ là khoảng cách).

Để biểu diễn độ đo giữa các đối tượng nêu trên ta có thể sử dụng ma trận độ đo khoảng cách được mô tả ở phần sau.

## 3. MA TRẬN KHOẢNG CÁCH

Một tập quan hệ được định nghĩa là tập các đối tượng có ràng buộc trong quan hệ đó và kích thước của tập quan hệ là số các đối tượng tham gia vào quan hệ này. Ví dụ lớp KHOA trong ví dụ 1 được xem là một tập quan hệ, kích thước của nó là số các thể hiện của lớp này. Mỗi tập quan hệ được kí hiệu là $R_m$, kích thước của quan hệ này là $|R_m|$.

Ma trận khoảng cách là ma trận xác định độ đo khoảng cách giữa các cặp đối tượng. Khoảng cách giữa mỗi cặp đối tượng được xác định như sau:

$$d(O_i, O_j) = \sum_{m=1}^{n} P_m * (\delta_{i,j,R_m} * \frac{1}{2} * |R_m| * s + \beta_{i,j,R_m} * \frac{1}{2} * L) * \omega_{i,j} \quad (1)$$

trong đó, $n$- số các tập quan hệ; $s$- kích thước đối tượng. Chúng ta giả sử rằng, trong hệ thống mọi đối tượng đều có kích thước như nhau và bằng $s$; $L$- tổng kích thước của tất cả các đối tượng và có giá trị $L = k * s$, với $k$ là tổng số các đối tượng trong hệ thống; $P_m$- xác suất mà tập quan hệ $R_m$ được hệ thống gom cụm, xác suất này có được từ phân bố xác suất của các mẫu truy vấn, nếu các mẫu xảy ra thường xuyên hơn thì xác suất tương ứng sẽ là lớn hơn. Chúng ta giả sử rằng xác suất của mọi tập quan hệ là giống nhau và bằng $\frac{1}{n}$.

Khi đó, khoảng cách $d(O_i, O_j)$ là tổng của tích xác suất và khoảng cách trung bình của các đối tượng được gom cụm trên mỗi quan hệ $R_m$, $R_m$ là quan hệ thứ $m$ hay category thứ $m$; $\delta_{i,j,R_m}$, $\beta_{i,j,R_m}$ và $\omega_{i,j}$ là các hàm đặt trưng có giá trị là 1 hoặc 0, được xác định như sau:

$$\delta_{i,j,R_m} = \begin{cases} 1 & \text{nếu } O_i \text{ và } O_j \text{ thuộc trong cùng tập quan hệ } R_m \\ 0 & \text{nếu } O_i \text{ và } O_j \text{ không thuộc trong cùng tập quan hệ } R_m \end{cases} \quad (2)$$

$$\beta_{i,j,R_m} = \begin{cases} 1 & \text{nếu } \delta_{i,j,R_m} = 0 \\ 0 & \text{nếu } \delta_{i,j,R_m} = 1 \end{cases} \tag{3}$$

$$\omega_{i,j,R_m} = \begin{cases} 1 & \text{nếu } O_i \neq O_j \\ 0 & \text{nếu } O_i \equiv O_j \end{cases} \tag{4}$$

**Định lý 1** . *Khoảng cách được định nghĩa trong công thức (1) là độ đo khoảng cách metric.*

*Chứng minh.*

1. $d(O_i, O_j) = 0$ khi và chỉ khi $O_i \equiv O_j$ (hai đối tượng đồng nhất)

   Ta có, $d(O_i, O_j) = \sum_{m=1}^{n} P_m \times (\delta_{i,j,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{i,j,R_m} \times \frac{1}{2} \times L) \times \omega_{i,j}$

   Vì $O_i \equiv O_j$ nên $\omega_{i,j} = 0$. Do đó, $d(O_i, O_j) = 0$

2. $d(O_i, O_j) = d(O_j, O_i)$ (tính giao hoán)

   $d(O_i, O_j) = \sum_{m=1}^{n} P_m \times (\delta_{i,j,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{i,j,R_m} \times \frac{1}{2} \times L) \times \omega_{i,j}$
   $= \sum_{m=1}^{n} P_m \times (\delta_{j,i,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{j,i,R_m} \times \frac{1}{2} \times L) \times \omega_{j,i}$
   $= d(O_j, O_i)$

3. $d(O_i, O_t) + d(O_t, O_j) \geq d(O_i, O_j)$ (bất đẳng thức tam giác)
   $d(O_i, O_t) + d(O_t, O_j) = \sum_{m=1}^{n} P_m \times (\delta_{i,t,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{i,t,R_m} \times \frac{1}{2} \times L) \times \omega_{i,t}$
   $+ \sum_{m=1}^{n} P_m \times (\delta_{t,j,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{t,j,R_m} \times \frac{1}{2} \times L) \times \omega_{t,j}$
   $= \sum_{m=1}^{n} P_m \times [(\delta_{i,t,R_m} + \delta_{t,j,R_m}) \times \frac{1}{2} \times |R_m| \times s + (\beta_{i,t,R_m} + \beta_{t,j,R_m}) \times \frac{1}{2} \times L] \times \omega_{i,j}$
   $\geq \sum_{m=1}^{n} P_m \times (\delta_{i,j,R_m} \times \frac{1}{2} \times |R_m| \times s + \beta_{i,j,R_m} \times \frac{1}{2} \times L) \times \omega_{i,j} = d(O_i, O_j)$

Từ 1, 2 và 3 ta có khoảng cách được định nghĩa trong công thức (1) là khoảng cách metric.

■

Bây giờ, giả sử chỉ có hai quan hệ instance-of và part-of tồn tại trong hệ thống, giả sử quan hệ instance-of được chỉ mục như $R_1$ và part-of được chỉ mục như $R_2$. Chúng ta chỉ ra cách tính khoảng cách giữa hai đối tượng áp dụng công thức (1) trên. Giả sử xác suất $P_1$ và $P_2$ là như nhau và bằng $\frac{1}{2}$; kích thước các đối tượng bằng nhau và bằng 1 ($s = 1$ và $L = k$); $C_h$ lớp chứa tập các đối tượng; $CO_h$ lớp chứa tập các đối tượng phức ($h = 1, 2$). Khi đó khoảng cách giữa hai đối tượng $O_i$ và $O_j$ được xác định như sau:

$$\begin{array}{lllll} & R_1 & & R_2 & \\ \text{- Trường hợp 1:} & P_1 \times (1 \times \frac{1}{2} \times |C_1|) & + & 0 \times \frac{1}{2} \times L) & \\ & +P_2 \times (0 \times \frac{1}{2} \times |CO_1| & + & 1 \times \frac{1}{2} \times L) & \\ & = \frac{1}{2} \times (\frac{1}{2} \times |C1| & + & \frac{1}{2} \times L) & \end{array}$$

nếu $O_i$ và $O_j$ trong cùng lớp $C_1(\delta_{i,j,R_1} = 1, \beta_{i,j,R_1} = 0)$ nhưng không cùng đối tượng $CO_1(\delta_{i,j,R_2} = 0, \beta_{i,j,R_2} = 1)$.

Tính tương tự cho các trường hợp 2, 3 và 4

$$\begin{array}{lllll} \text{- Trường hợp 2:} & \frac{1}{2} \times (\frac{1}{2} \times L & + & \frac{1}{2} \times |CO_1|) & \end{array}$$

nếu $O_i$ và $O_j$ không cùng lớp $C_1$ nhưng cùng đối tượng phức $CO_1$

$$\begin{array}{lllll} \text{- Trường hợp 3:} & \frac{1}{2} \times (\frac{1}{2} \times |C_2| & + & \frac{1}{2} \times |CO_2|) & \end{array}$$

nếu $O_i$ và $O_j$ cùng lớp $C_2$ và cùng đối tượng phức $CO_2$

$$\begin{array}{lllll} \text{- Trường hợp 4:} & \frac{1}{2} \times (\frac{1}{2} \times L & + & \frac{1}{2} \times L) & \end{array}$$

nếu $O_i$ và $O_j$ không cùng lớp và không cùng đối tượng phức

**Ví dụ 3.** Xét cơ sở dữ liệu được cho trong ví dụ 1. Trong lược đồ này chỉ tồn tại hai loại quan hệ là instance-of và part-of. Khi đó, khoảng cách giữa hai đối tượng $O_i$ và $O_j$ trong lớp KHOA sẽ là:

$$\frac{1}{2} \times (1 \times \frac{1}{2} \times |KHOA| + 0 \times \frac{1}{2} \times L) + \frac{1}{2} \times (0 \times \frac{1}{2} \times |KHOA| + 1 \times \frac{1}{2} \times L) = \frac{1}{2} \times (\frac{1}{2} \times |KHOA| + \frac{1}{2} \times L)$$

nếu chúng không có tham chiếu đệ quy. Nếu $O_i$ và $O_j$ được tham chiếu đệ quy thì khoảng cách sẽ là:

$$\frac{1}{2} \times (\frac{1}{2} \times |KHOA| + \frac{1}{2} \times |CO_{khoa}|).$$

Ta có thể thấy rằng khoảng cách của các trường hợp trên luôn thỏa mãn các bất đẳng thức sau:

$d$(Trường hợp 3) < $d$(Trường hợp 1) < $d$(Trường hợp 4) và

$d$(Trường hợp 3) < $d$(Trường hợp 2) < $d$(Trường hợp 4),

trong đó, $d$(Trường hợp 1) $= \frac{1}{2} \times (\frac{1}{2} \times |C_1| + \frac{1}{2} \times L)$,

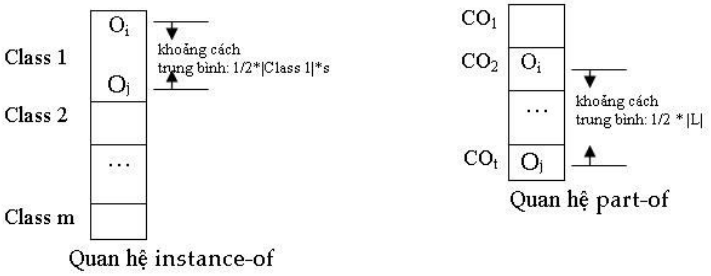
$d$(Trường hợp 2) $= \frac{1}{2} \times (\frac{1}{2} \times L + \frac{1}{2} \times |CO_1|)$,

$d$(Trường hợp 3) $= \frac{1}{2} \times (\frac{1}{2} \times |C_2| + \frac{1}{2} \times |CO_2|)$,

$d$(Trường hợp 4) $= \frac{1}{2} \times (\frac{1}{2} \times L + 12 \times L)$.

vì $|R_i| < L$ thỏa mãn cho mọi tập quan hệ trong hệ thống. Điều này chứng tỏ nó thỏa mãn ràng buộc 1 nêu trên, vì hai đối tượng có liên quan chặt chẽ hơn thì có khoảng cách nhỏ hơn. Tuy nhiên, khoảng cách của Trường hợp 1 và Trường hợp 2 là không thể so sánh, vì nó phụ

thuộc vào kính thước của tập quan hệ $|C_1|$ và $|CO_1|$. Điều này cũng thỏa ràng buộc 2 vì do quan hệ liên quan không thể so sánh được.




*Hình 2.* Cách tính khoảng cách trung bình giữa hai đối tượng

Mục đích của chúng ta là làm rõ mức độ các mối quan hệ giữa các đối tượng được lượng hóa trong công thức tính khoảng cách (1). Nếu hai đối tượng $O_i$, $O_j$ thuộc trong cùng lớp và chúng ta gom cụm chúng theo quan hệ instance-of, thì khoảng cách sẽ là trung bình kích thước của tất cả các đối tượng trong tập quan hệ. Hình 2 giải thích cách tính khoảng cách giữa $O_i$ và $O_j$ đạt được trong trường hợp 1.

Ma trận khoảng cách của $k$ đối tượng $\{O_1, ..., O_k\}$ là ma trận có kích thước $k \times k$ được định nghĩa như sau:

$$M = (d(O_i, O_j))_{k \times k}$$

Ma trận $M$ đối xứng qua đường chéo chính và các phần tử trên đường chéo chính có giá trị bằng 0. Do đó, để tính ma trận $M$ ta chỉ cần tính các phần tử $M(i, j)$ với $j > i$.

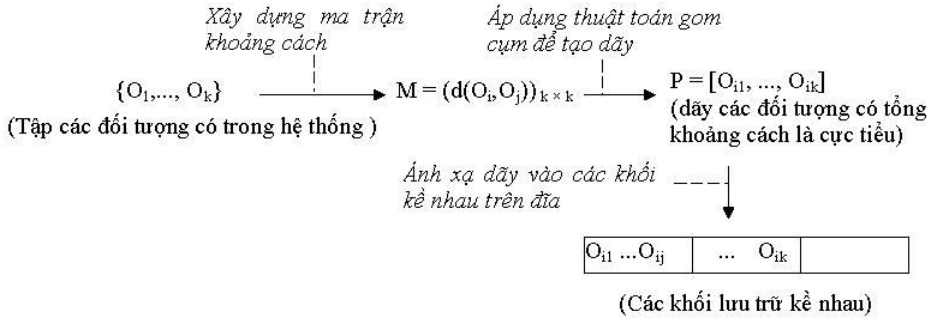
## 4. THUẬT TOÁN GOM CỤM ĐỐI TƯỢNG

Vì cần phải sắp đặt tất cả các đối tượng vào bộ nhớ lưu trữ thứ cấp, nên cách tốt nhất là sắp xếp tất cả đối tượng của hệ thống vào cấu trúc dữ liệu một chiều, sau đó ánh xạ cấu trúc dữ liệu một chiều này vào bộ nhớ lưu trữ thứ cấp. Do đó, thuật toán của chúng tôi có kết quả đầu ra là dãy gồm tất cả các đối tượng có trong hệ thống, sao cho tổng khoảng cách giữa các đối tượng trong dãy là đạt cực tiểu. Với thuật toán này thì tất cả các tính chất của quan hệ bội vẫn được bảo toàn. Quá trình xử lý của chúng tôi có thể được khái quát hóa trong Hình 3.

Sau khi ma trận khoảng cách $M = (d(O_i, O_j))_{k \times k}$ được xây dựng bằng cách tính khoảng cách giữa mỗi cặp đối tượng, thuật toán gom cụm đối tượng được sử dụng để sắp xếp các đối tượng vào dãy. Đầu vào của thuật toán là ma trận khoảng cách $M$ và $W = \{O_1, O_2, ..., O_k\}$ là tập các đối tượng có trong hệ thống, mỗi đối tượng trong đó có thể tham gia vào hơn một loại tập quan hệ. Để đơn giản, chúng ta có thể xem $M$ là 1 ma trận trọng số (ma trận kề), W là tập các đỉnh của đồ thị liên thông mạnh[1]. Xuất phát từ ý tưởng này, thuật toán đề

[1]là đồ thị vô hướng mà giữa hai đỉnh bất kỳ của nó luôn có cạnh nối.

xuất sẽ xác định một đường đi qua tất cả các đỉnh của đồ thị một đỉnh đúng một lần, sao cho tổng khoảng cách giữa mỗi hai đỉnh (đối tượng) trong đường đi (dãy) là cực tiểu.




*Hình 3.* Tiến trình thực hiện gom cụm các đối tượng

**Thuật toán**

**Input**

- Ma trận khoảng cách $M = (d(O_i, O_j))_{k\times k} = (d_{ij})_{k\times k}$
- Tập tất cả các đối tượng cần được gom cụm $W = \{O_1, O_2, ..., O_k\}$.

**Output**

Dãy gồm tất cả các đối tượng $\{O_1, O_2, ..., O_k\}$, sao cho tổng khoảng cách giữa các cặp đối tượng trong dãy là cực tiểu.

**Method**

(1) $t := O_i$; {chọn ngẫu nhiên một đối tượng $O_i \in W$ }

(2) $h := 1$;

(3) $p[h] := t$;

{mảng $p$ lưu trữ dãy các đối tượng trong $W$, thỏa mãn điều kiện là tổng khoảng cách giữa mỗi cặp đối tượng trong dãy $p$ cực tiểu}

(4) $W := W - \{t\}$; {loại bỏ đối tượng $t$ đã xét ra khỏi $W$}

(5) **While** $(h \leq k)$ **and** $(W \neq \varnothing)$ **do**

**begin**

(6) $d := \infty$;

(7) **for each** $O_j \in W$ **do**

(8) **if** $(d[t, O_j] < d)$ **and** $(d[t, O_j] > 0)$ **then**

**begin**

(9) $d := d[t, O_j]$;

(10) $p[h] := O_j$;

**end;**

{khi kết thúc vòng lặp này ta có $d$ là khoảng cách bé nhất từ đối tượng $t$ đến đối tượng $O_j$, và $O_j$ sẽ là đối tượng được chọn ở bước duyệt tiếp theo}

(11) $t := p[h]$; {$O_j$ là đối tượng được chọn cho bước duyệt tiếp theo}

(12) $h := h + 1$;

(13) $W := W - \{t\}$;

**end;**

(14) **return** $P$;

Độ phức tạp tính toán của giải thuật phụ thuộc vào hai vòng lặp (5) và (7) lồng nhau, mỗi vòng thực hiện tối đa $k$ lần, nên ta có số lần thực hiện vòng lặp (7) là $O(k^2)$. Do vậy, giải thuật có độ phức tạp là $O(k^2)$. Kết quả đầu ra của thuật toán phụ thuộc vào việc chọn đối tượng $O_i$ ngẫu nhiên trong $W$ ban đầu.

**Ví dụ 4**. Xét cơ sở dữ liệu hướng đối tượng được cho ở Hình 1, ta có:

$k = 6$ (có tất cả 6 đối tượng),

$n = 2$ (hai quan hệ là instance-of và part-of),

$P_1 = P_2 = \dfrac{1}{2}$.

Giả sử kích thước của mỗi đối tượng là bằng nhau và bằng 1 ($s = 1$), khi đó: $L = k \times s = 6$.

Bước 1. Tính ma trận khoảng cách: $M = (d(O_i, O_j))_{6\times 6}$. Vì ma trận $M$ đối xứng qua đường chéo chính, nên ta chỉ cần tính các phần tử $d(O_i, O_j)$ với $j > i$. Ma trận khoảng cách $M$ tính được dựa trên các đối tượng đã cho như sau:

| | $O_1$ | $O_2$ | $O_3$ | $O_4$ | $O_5$ | $O_6$ |
|---|---|---|---|---|---|---|
| $O_1$ | 0.00 | 1.50 | 2.25 | 3.00 | 3.00 | 3.00 |
| $O_2$ | | 0.00 | 2.25 | 2.25 | 2.25 | 3.00 |
| $O_3$ | | | 0.00 | 2.25 | 2.25 | 3.00 |
| $O_4$ | | | | 0.00 | 2.25 | 2.25 |
| $O_5$ | | | | | 0.00 | 2.25 |
| $O_6$ | | | | | | 0.00 |

Bước 2. Áp dụng thuật toán gom cụm trên, với đối tượng được chọn xuất phát là $O_3$, ta có dãy đối tượng với tổng khoảng cách cực tiểu là 10.5 như sau: $O_3, O_1, O_2, O_4, O_5, O_6$.

## 5. KẾT LUẬN

Dựa vào ý tưởng sử dụng ma trận khoảng cách, chúng tôi đã xây dựng công thức tính khoảng cách tổng quát giữa mỗi cặp đối tượng có trong hệ thống và chứng minh được độ đo khoảng cách này là metric, từ đó đề xuất một thuật toán tạo dãy chứa tất cả các đối tượng trong hệ thống dựa trên ma trận khoảng cách, sao cho tổng khoảng cách của các đối tượng trong dãy đạt cực tiểu. Từ đó, ánh xạ dãy các đối tượng này vào bộ nhớ lưu trữ thứ cấp. Với độ phức tạp tính toán là $O(k^2)$ và trong một số trường hợp, thuật toán có thể không cho lời giải tối ưu. Tuy nhiên, với số đối tượng lớn và các đối tượng tham gia vào nhiều mối quan hệ khác nhau thì thuật toán này vẫn áp dụng được một cách hiệu quả.

Thuật toán gom cụm đã đề xuất luôn cho lời giải tối ưu và có độ phức tạp tốt hơn so với các thuật toán đã đề xuất trong [4] và [6]. Độ phức tạp của thuật toán trong [6] là $O(k^3)$, trong [4] là $O(k^2)$, nhưng nhược điểm của thuật toán đề xuất trong [4] là tại mỗi bước lặp, chọn ngẫu nhiên một đối tượng chưa được xử lý rồi chèn vào dãy $p$ tại một vị trí đặt biệt có khoảng cách mở rộng đạt cực tiểu. Do đó, dãy $p$ thu được có thể có tổng khoảng cách có thể

không đạt cực tiểu. Ngược lại, thuật toán nêu trên duyệt tất cả các đối tượng chưa được xử lý để tìm ra khoảng cách cực tiểu.

Phần ánh xạ dãy các đối tượng vào các khối trên bộ nhớ thứ cấp sẽ được trình bày trong những nghiên cứu tiếp theo.

## TÀI LIỆU THAM KHẢO

[1] Adrian Darabant, Alina Câmpan, Grigor Moldovan, Horea Grebla, Clustering techniques: a new approach in horizontal fragmentation of classes with complex attributes and methods in Object-Oriented DataBases, *ICTAMI 2004*, Thessaloniki, Greece, 2004.

[2] Gabriela Serban, Alina Campan, Hierarchical Adaptive Clustering, *Informatica* **19** (1) (2008) 101–102.

[3] LingWang, Liefeng Bo, Licheng Jiao, "A modified K-Means clustering with Density-Sensitive Distance Metric", Institute of Intelligent Information processing, Xidian University Xi'an 710071, China, 2008.

[4] M. Manolis, Tsangaris, and Jeffery F. Naughton, A stochastic approach for clustering in object bases, *ACM SIGMOD Intl. Conf. On Management of Data*, Denver, Colorado, 1991.

[5] Patrick Valduriez, Setrag Khoshafian, George Copeland, Implementation techniques of complex objects, *Proceeding of Twelfth International Conference*, Kyoto, 1996.

[6] Steven Yi-Cheng Tu, Daniel J.Bueher, Clustering objects for OODBS in a multiple relationship environment, *Proceeding of Pan Pacific Conference on Information Systems*, 1993 (p.128).

[7] Sophie Chabridon, Jen-Chyi Liao, Yichen Ma, Le Gruenwald, Clustering techniques for object-oriented database systems, *IEEE Compcon Spring*, February 1993 (232–242).

[8] Trigoni, Agathoniki, "Semantic Optimization of OQL Queries", Technical Report, Number 547, University of Cambridge, Computer Laboratory, UCAM-CL-TR-547, ISSN 1476-2986, October, 2002.

[9] Vlad S. I. Wietrzyk, Mehmet A. Orgun, *Clustering techniques for minimizing object access time*, Springer, Volume 1475, (2004) 236–247.

*Nhận bài ngày 15 - 5 - 2009*
*Nhận lại sau sửa ngày 7 - 10 -2009*