LỜI GIẢI TỐI ƯU VÀ TẬP SINH TRÊN MẠNG SUY DIỄN

Nguyễn Hữu Anh – Đỗ Văn Nhơn
Trường Đại học Khoa học Tự nhiên
(Bài nhận ngày 10 tháng 01 năm 2001)

TÓM TẮT: *Mạng suy diễn là một mô hình biểu diễn tri thức đã được đề cập đến trong [1] và [2]. Một số thuật toán mô phỏng một quá trình tư duy giải toán được phát biểu và chứng minh để có thể áp dụng giải tự động các bài toán trên máy tính. Trong bài báo nầy chúng tôi khảo sát vấn đề về lời giải tối ưu trên mạng suy diễn, và khảo sát khái niệm tập hợp sinh trên mạng suy diễn để làm cơ sở cho việc bổ sung giả thiết cũng như việc tìm ra tập các thuộc tính đặc trưng cho mạng suy diễn.*

Giới thiệu

Trong [2] các tác giả đã giải quyết vấn đề tìm một lời giải trên mạng suy diễn. Vấn đề ta quan tâm ở đây là việc tìm một lời giải tốt nhất (hay lời giải tối ưu) và tìm các thuộc tính đặc trưng cho mạng suy diễn cũng như tìm thuộc tính bổ sung cho giả thiết của bài toán trong trường hợp nó không có lời giải. Quan niệm về lời giải tốt nhất phụ thuộc vào việc ta quan tâm đến mục tiêu tối ưu nào đối với lời giải. Trong nhiều trường hợp áp dụng cụ thể như hệ suy diễn tính toán hay hệ suy diễn trên các phản ứng hóa học, mỗi luật suy diễn có một tham số tương ứng đại diễn cho độ phức tạp của luật suy diễn hay chi phí. Những tham số nầy đóng vai trò quan trọng trong tính hiệu quả của lời giải. Chúng ta sẽ xem xét các mạng suy diễn có trọng số, trong đó ứng với mỗi luật suy diễn ta có một trọng số dương tương ứng, và lời giải tối ưu sẽ được đề cập đến theo nghĩa là tổng trọng số thấp nhất. Lời giải ngắn nhất chính là lời giải tối ưu trong trường hợp trọng số của các luật dẫn đều là 1.

Khái niệm tập hợp sinh và phương pháp tìm một tập hợp sinh trên một mạng suy diễn là cơ sở cho việc phát triển các thuật toán giải quyết vấn đề kiểm định giả thiết của bài toán suy diễn. Hơn nữa việc khảo sát về tập hợp sinh cũng là một vấn đề được đặt ra một cách tự nhiên trên mạng suy diễn nhằm tìm ra một tập hợp tối thiểu các thuộc tính và các luật suy diễn sinh ra tất cả các thuộc tính khác.

Mạng Suy diễn có Trọng số và lời giải tối ưu

1. Định nghĩa và ký hiệu

- **Định nghĩa 1**: Ta gọi một *mạng suy diễn có trọng số* là một mô hình (A, D, w) bao gồm: một tập hợp các thuộc tính A, một tập hợp các luật suy diễn D, và một hàm trọng số dương $w : D \to \mathbf{R}^+$. Mỗi luật suy diễn r thuộc D có dạng $r : u \Rightarrow v$, với u và v là các tập hợp con khác rỗng và rời nhau của A. Ta gọi u là phần giả thiết của luật r và ký hiệu là *hypothesis*(r). Tập v được gọi là phần kết luận của luật r và ký hiệu là *goal*(r). Tập hợp *attr*(r) = *hypothesis*(r) $\cup$ *goal*(r) được gọi là tập hợp các thuộc tính trong luật r. Mạng suy diễn có trọng số sẽ được viết tắt là **MSDT**.

Ghi chú:

- Tập hợp D các luật suy diễn có thể viết dưới dạng một tập hợp các quan hệ suy diễn mà mỗi quan hệ suy diễn xác định các luật suy diễn có trọng số bằng nhau.
- Trong một MSDT (A, D, w) ta có (A, D) là một MSD.

Ví dụ 1: Gọi *A* là tập hợp gồm 3 cạnh của một tam giác (a, b, và c), 3 góc trong (A, B, và C), nửa chu vi p, diện tích S, 3 đường cao (ha, hb, và hc), và các yếu tố khác của tam giác: *A* = {A, B, C, a, b, c, p, S, ha, hb, hc, ...}. Tập các quan hệ suy diễn thể hiện bởi các công thức liệt kê trong tập hợp sau đây:

D = { f1:A + B + C = 180; f2:a/sin(A) = b/sin(B); f3:b/sin(B) = c/sin(C);
f4:a/sin(A) = c/sin(C); f5:2*p = a + b + c; f6:S = a*ha/2; f7:S = b * hb / 2;
f8:S = c*hc/2; f9:S = sqrt(p*(p-a)*(p-b)*(p-c));
f10:ha = b*sin(C); f11:hb = a*sin(C); f12:hc = b*sin(A) }.

Giả sử các phép toán +, -, * và / được đặt cho trọng số là 1, phép tính căn bậc 2 có trọng số là một hằng số dương c1, và các tính toán hàm lượng giác có trọng số là một hằng số dương c2; trong đó c1 >> 1 và c2 >> 1. Như thế các quan hệ suy diễn có trọng số tương ứng như sau:

w(f1) = 2; w(f2) = w(f3) = w(f4) = 2*c2 + 2;
w(f5) = 3; w(f6) = w(f7) = w(f8) = 2;
w(f9) = c1 + 6; w(f10) = w(f11) = w(f12) = c2 + 1;

Khi đó ta có (*A*, *D*, w) là một mạng suy diễn có trọng số.

- **Định nghĩa 2**: Giả sử (*A*, *D*, w) là một MSDT. Cho $S = \{f_1,...,f_k\}$ là một dãy các luật suy diễn và *H* là một tập hợp các thuộc tính. Đặt
  $S_1(H) = H \cup goal(f_1)$ nếu $hypothesis(f_1) \subseteq H$, ngược lại $S_1(H) = H$.
  $S_i(H) = S_{i-1}(H) \cup goal(f_i)$ nếu $hypothesis\ (f_i) \subseteq S_{i-1}(H)$, ngược lại $S_i(H) = S_{i-1}(H)$.
  $S(H) = S_k(H)$. $w(S) = w(f_1) + w(f_2) + ... + w(f_k)$. Ta gọi w(*S*) là trọng số của *S*.
  Cho một bài toán $H \to G$. Dãy các luật suy diễn *S* được gọi là một lời giải tối ưu của bài toán khi nó thỏa mãn các điều kiện sau đây:
  (1) *S* là một lời giải của bài toán $H \to G$, nghĩa là $S(H) \supset G$.
  (2) $w(S) = \min \{w(S') \mid S'$ là một lời giải của bài toán $H \to G\}$

2. Tìm lời giải tối ưu

Trong mục nầy ta sẽ xem xét vấn đề tìm lời giải tối ưu cho bài toán $H \to G$ trên một MSDT (*A*, *D*, w). Cách tiếp cận để giải quyết vấn đề nầy là dựa trên thuật giải A^* ([4]). Để có thể áp dụng thuật giải nầy chúng ta cần có một biểu diễn thích hợp cho không gian trạng thái của bài toán cũng như cho yêu cầu của bài toán.

- Không gian trạng thái của bài toán: Xét bài toán $H \to G$ trên một MSDT (*A*, *D*, w). Với mỗi luật r mà ta có thể áp dụng trên *H* để suy ra những thuộc tính mới (nghĩa là *hypothesis*(r) bao hàm trong *H* nhưng *goal*(r) thì không bao hàm trong *H*) sẽ dẫn tới một tập thuộc tính mới $H' = H \cup goal(r)$.Ta nói rằng r là một *cạnh* nối từ *đỉnh H* đến *đỉnh H'*. Như thế, tập hợp gồm tất cả các tập con (hay đỉnh) *H'* của *A* sao cho có một dãy *S* gồm các luật (hay cạnh) thỏa mãn điều kiện $H' = S(H)$, cùng với các cạnh trong các dãy *S* sẽ cho ta một không gian trạng thái của bài toán có dạng đồ thị. Hơn nữa đồ thị nầy có trọng số được định nghĩa bởi: trọng số của cạnh r (tức là một luật suy diễn) là w(r). Đồ thị có trọng số nầy sẽ được ký hiệu là Graph($H \to G$). Như đã biết trong lý thuyết đồ thị, độ dài của một lộ trình trên một đồ thị có trọng số là tổng trọng số của tất cả các cạnh trên lộ trình. Từ cách xây dựng đồ thị của bài toán ta có mệnh đề sau đây:

Mệnh đề 1:

(1) Một dãy *S* gồm các luật là một lời giải của bài toán $H \to G$ khi và chỉ khi *S* là một lộ trình trên đồ thị Graph($H \to G$) nối từ *H* đến *S*(*H*) và $S(H) \supset G$.

(2) Độ dài của một lộ trình S trên đồ thị Graph($H \rightarrow G$) là w(S), trọng số của danh sách luật S trên MSDT (A, D, w).

Từ mệnh đề nầy, việc tìm lời giải tối ưu cho bài toán $H \rightarrow G$ tương đương với việc tìm một đường đi ngắn nhất trên đồ thị Graph($H \rightarrow G$) từ H đến một đỉnh mục tiêu H' thỏa điều kiện H' chứa G. Đối với mỗi đỉnh N trên đồ thị, đặt $h(N)=\min\{w(r) \mid hypothesis(r) \subset N\}$. Giá trị h(N) nầy có thể xem là một ước lượng cho lộ trình từ N đến một đỉnh mục tiêu. Từ đó chúng ta có thể viết thuật toán tìm lời giải tối ưu cho bài toán như sau:

<u>**Thuật toán 1**</u>

```
Bước 1: Khởi tạo trạng thái xuất phát.
        Open ← {H}; Close ← {};
        g(H) ← 0;       // độ dài lộ trình đến H là 0
        f(H) ← h(H);   // độ dài lộ trình ước tính từ H đến mục tiêu là h(H)
        found ← false;        // biến kiểm tra quá trình tìm lời giải
Bước 2: Thực hiện quá trình lặp để tìm lời giải tối ưu.
    While (Open ≠ {}) do Begin
        Bước 2.1: Chọn một đỉnh N trong Open với ước tính đường đi f nhỏ nhất.
        Bước 2.2: Chuyển N từ danh sách Open sang danh sách Close.
        Bước 2.3: if (N là một mục tiêu) then Begin
                      Found ← true;
                      Break;            // Kết thúc quá trình lặp
                      End
        Bước 2.4: else
                  Duyệt qua các đỉnh kế S của N (tức là có cung r nối N và S) mà S ∉
                  Close, ứng với mỗi S ta xét các trường hợp sau:
                  1. S ∉ Open: Tính g(S) = g(N) + w(r); f(S) = g(S) + h(S);
                     Bổ sung S vào Open;
                  2. S ∈ Open:
                     If g(N) + w(r) < g(S) then Begin
                        g(S) ← g(N) + w(r);  f(S) ← g(S) + h(S);
                        Cập nhật thông tin về đỉnh kế trước của S trên lộ trình;
                        End
        End   // Kết thúc vòng lặp while
Bước 3: Kiểm tra kết quả việc tìm kiếm.
        If Found then Kết quả là tìm được lời giải tối ưu và thiết lập lời giải
        Else Kết quả là bài toán không có lời giải.
```

<u>**Định lý 1**</u> Thuật toán trên cho một lời giải tối ưu của bài toán và có độ phức tạp là $O(|A|^2.|D|^2)$.

<u>Ví dụ 2</u>: Giả sử ta có mạng suy diễn có trọng số (A, D, w) như trong ví dụ 1.

Xét bài toán $H \rightarrow G$ với H = {a, b, B} và G = {S}. Trên mạng suy diễn (A, D), nếu áp dụng thuật toán suy diễn tiến được trình bày trong [2] ta có thể tìm được một lời giải S = {f2, f1, f3, f5, f9}. Trên mạng suy diễn có trọng số (A, D, w) lời giải S có trọng số là w(S) = 4*c2 + c1 + 15. Áp dụng thuật toán 1 để tìm lời giải trên mạng (A, D, w) ta có thể tìm được một lời giải tối ưu S' = {f2, f1, f10, f6} với trọng số là w(S') = 3*c2 + 7.

Tập hợp sinh

1. Khái niệm tập hợp sinh

- **Định nghĩa 3**: Cho (A, D) là một mạng suy diễn. Một tập thuộc tính $S \subset A$ được gọi là một *tập hợp sinh* của mạng suy diễn khi ta có bao đóng của S trên mạng là A, nghĩa là $\overline{S} = A$.

Từ định nghĩa về tập hợp sinh ở trên ta có các tính chất sau:

(1) Nếu S là một tập hợp sinh trong một mạng suy diễn và $S \subset T$ thì T cũng là một tập hợp sinh.

(2) Nếu S là một tập hợp sinh trong mạng suy diễn (A, D) và D' là một tập hợp mở rộng của D, nghĩa là $D \subset D'$, thì ta cũng có S là một tập hợp sinh của mạng suy diễn (A, D').

2. Tìm tập hợp sinh

Trong mục nầy sẽ trình bày cách tìm một tập hợp sinh không tầm thường với số phần tử càng ít càng tốt của mạng suy diễn. Ta có thể tìm một tập hợp sinh bằng phương pháp thử dần. Tuy nhiên chúng ta sẽ xây dựng một thuật toán tốt hơn bằng cách thiết lập một "biểu đồ (hay đồ thị) phân cấp" của một mạng suy diễn.

- **Định nghĩa 4**: Cho một mạng suy diễn (A, D). Ta xây dựng một đồ thị định hướng Graph(A, D) như sau:

  (1) Tập hợp đỉnh gồm tất cả các thuộc tính và tất cả các luật suy diễn, tức là $A \cup D$.

  (2) Ứng với mỗi luật r : *hypothesis*(r) $\rightarrow$ *goal*(r) ta thiết lập một tập hợp *edges*(r) gồm tất cả các cung định hướng (x, r) và (r, y) thỏa $x \in$ *hypothesis*(r) và $y \in$ *goal*(r) một cách tương ứng. Tập hợp các cạnh của đồ thị Graph(A, D) là hợp của tất cả các tập hợp *edges*(r) với r chạy trong tập D.

  Trong trường hợp trên đồ thị Graph(A, D) ta có $deg_{in}(x) \leq 1$ với mọi $x \in A$, thì ta có thể hiểu ngầm các đỉnh thuộc D và xét một đồ thị thu gọn $Graph_D(A)$ gồm:

  (1) Tập hợp đỉnh là tập hợp các thuộc tính A.

  (2) Tập hợp cạnh gồm tất cả các cung (x,y) thỏa mãn điều kiện: Có (duy nhất) một luật r sao cho *goal*(r) = y và $x \in$ *hypotheis*(r).

- **Định nghĩa 5**: Ta gọi một đồ thị định hướng đơn giản không có chu trình là một *biểu đồ phân cấp* (hay một đồ thị phân cấp). Đỉnh x của đồ thị mà không có cung hướng tới sẽ được gọi là đỉnh mức 0, và ta viết *level*(x) = 0. Đối với các đỉnh x khác, ta định nghĩa một cách qui nạp mức của nó và mức của đỉnh nầy có thể được cho bởi

  $$level(x) = \max \{level(a) \mid \text{có cung nối a với x}\} + 1$$

  Như thế tập hợp đỉnh của biểu đồ phân cấp được sắp xếp thành các mức, với mức 0 của biểu đồ là tập hợp tất cả các đỉnh mức 0, mức 1 của biểu đồ là tập hợp tất cả các đỉnh mức 1, v...v...

Ví dụ 3: Đồ thị trong Hình 1 là một biểu đồ phân cấp với 5 mức là:

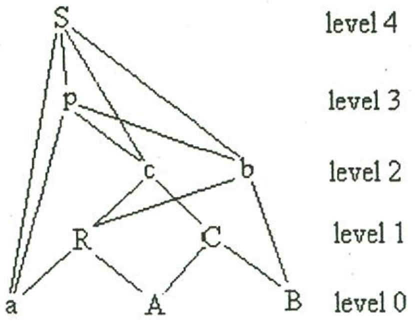
$Level_0 = \{a, A, B\}$, nghĩa là *level*(a) = *level*(A) = *level*(B) = 0.

$Level_1 = \{R, C\}$, nghĩa là *level*(R) = *level*(C) = 1.

$Level_2 = \{c, b\}$, nghĩa là *level*(c) = *level*(b) = 2.

$Level_3 = \{p\}$, nghĩa là *level*(p) = 3.

$Level_4 = \{S\}$, nghĩa là *level*(S) = 4.



Hình 1: Một biểu đồ phân cấp gồm 5 mức.

Từ các định nghĩa trên ta có thể dễ dàng chứng minh mệnh đề và định lý dưới đây:

Mệnh đề 2: Cho mạng suy diễn (A, D). Giả sử đồ thị Graph(A, D) có đồ thị thu gọn Graph$_D(A)$. Khi ấy, nếu Graph$_D(A)$ là một đồ thị phân cấp thì tập hợp S = Level$_0$ gồm tất cả các đỉnh mức 0 sẽ cho ta một tập hợp sinh của mạng suy diễn. Hơn nữa trong trường hợp nầy ta còn có:

(1) S là tập hợp sinh nhỏ nhất trên mạng suy diễn.

(2) Tập D là tập hợp luật tối thiểu để Level$_0$ sinh ra A. Nói một cách khác, nếu D' là một tập hợp con thật sự của D thì S không phải là tập hợp sinh trên mạng suy diễn (A, D').

Định lý 2: Cho mạng suy diễn (A, D). ta có:

(1) $S \subset A$ là một tập hợp sinh trên mạng suy diễn khi và chỉ khi có một tập luật $D' \subset D$ sao cho Graph(A, D') là một đồ thị phân cấp và S chứa tập hợp các đỉnh mức 0 của đồ thị nầy.

(2) Tồn tại một tập luật $D' \subset D$ sao cho Graph(A, D') là một đồ thị phân cấp.

Định lý trên cho ta một cách tìm một tập hợp sinh trên mạng suy diễn (A, D) bằng thuật toán sau đây:

Thuật toán 2: Tìm một tập hợp sinh S trong mạng suy diễn (A, D) bằng cách xây dựng một mạng con (A', D') với $A' = A$ và có Graph(A', D') là một biểu đồ phân cấp.

Bước 1: $A' \leftarrow \{\}$; $D' \leftarrow \{\}$; $S \leftarrow \{\}$;

Bước 2: For $r \in D$ do

If not $(attr(r) \subset A')$ then

$A' \leftarrow A' \cup attr(r)$; $D' \leftarrow D' \cup \{r\}$; và

Thực hiện việc cập nhật A', D' và S theo 2 trường hợp như sau:

- Trường hợp 1: goal(r) $\notin A'$
  $S \leftarrow S \cup (hypothesis(r) - A')$;
- Trường hợp 2: goal(r) $\in A'$
  Loại r' (nếu có) trong D' mà $goal(r') = goal(r)$;
  Loại một số luật suy ra các $x \in hypothesis(r)$ để bảo đảm tính phân cấp của biểu đồ và cập nhật S.

Bước 3: $S \leftarrow S \cup (A - A')$; $A' \leftarrow A$

3. Bổ sung giả thiết cho bài toán suy diễn

Trong mục nầy chúng ta xem xét việc bổ sung giả thiết cho bài toán $H \rightarrow G$ trên một mạng suy diễn (A, D) trong trường hợp bài toán không giải được. Ý tưởng chính ở đây là tiến hành một quá trình xây dựng một biểu đồ phân cấp với tập hợp đỉnh chứa G và ưu tiên cho việc đặt các phần tử của H ở mức 0. Thuật toán cơ bản dưới đây cho ta một cách để tìm một tập thuộc tính H' sao cho $H \cap H' = \varnothing$ và bài toán $(H \cup H') \rightarrow G$ là giải được trên mạng suy diễn.

Thuật toán 3: Cho mạng suy diễn (A, D) và bài toán $H \rightarrow G$ không giải được (không có lời giải). Tìm H' sao cho $H \cap H' = \varnothing$ và bài toán $(H \cup H') \rightarrow G$ là giải được.

Bước 1: $A' \leftarrow H$; $D' \leftarrow \{\}$; $G \leftarrow G - A'$;

Bước 2: while $(G \neq \{\}$ and $D \neq \{\})$ do

2.1: Lấy ra một r từ D và cập nhật D.

2.2: if *hypothesis*(r) $\cap G \neq \{\}$ or *attr*(r) $\subseteq A'$ then Bỏ qua r

2.3: else Thêm r vào D' và bổ sung *attr*(r) vào A'và trong trường hợp goal(r) $\in A'$ thì loại ra r' từ D' (nếu có) thỏa *goal*(r') = *goal*(r) và loại một số luật suy ra các x $\in$ *hypothesis*(r) để bảo đảm tính phân cấp của biểu đồ.

2.4: $G \leftarrow G - A'$

Bước 3: if $G \neq \{\}$ then Kết thúc với kết luận:Vấn đề bổ sung giả thiết là không giải quyết được

Bước 4: else (Trong trường hợp nầy Graph(A',D') có biểu đồ phân cấp tương ứng là $\text{Graph}_{D'}A'$) Gọi L_0 là mức 0 của biểu đồ $\text{Graph}_{D'}A'$. Đặt: $H' \leftarrow L_0 - H$.

Định lý 3: Thuật toán tìm sự bổ sung giả thiết cho bài suy diễn được nêu trên là đúng và có độ phức tạp là $O(|A|.|D|)$.

Ghi chú: Chúng ta có thể sử dụng thêm các qui tắc heuristic trong bước 2 của thuật toán trên nhằm giảm thiểu số thuộc tính ở mức 0 và/hay để ưu tiên chọn lựa các thuộc tính được ưu tiên cho việc dùng bổ sung giả thiết của bài toán.

Kết luận

Các khảo sát về lời giải tối ưu trên mạng suy diễn có trọng số và về tập hợp sinh của mạng suy diễn đã góp phần giải quyết đầy đủ các vấn đề cơ bản trên mạng suy diễn làm cho mô hình mạng suy diễn có thể được áp dụng một cách hiệu quả hơn trong việc thiết kế và cài đặt các hệ giải toán tự động cũng như các hệ hỗ trợ học giải toán trên máy tính. Chúng không chỉ tìm cho ta kết quả mà còn có khả năng cung cấp một lời giải tốt và tự nhiên phù hợp với cách nghĩ và cách trình bày lời giải của con người. Đây là một trong những ưu điểm của mô hình so với các phương pháp được trình bày trong [3]. Ngoài ra, các kết quả và phương pháp được nêu lên ở đây có thể được mở rộng cho mạng các đối tượng và áp dụng vào các hệ giải toán phức tạp hơn như [5].

OPTIMAL SOLUTION AND GENERATED SET ON DEDUCTION NET

Nguyen Huu Anh – Do Van Nhon

Abstract: *Deduction Net is a model, which was proposed in [1] and [2]. It can be used to represent knowledge. Some algorithms that imitate the thinking process of human being were given and they can be used to solve problems on computers automatically. In this paper we consider the problem of finding an optimal solution on deduction nets and the concept "generated set", which is useful for solving the hypothesis completion problem as well as finding a set of attributes generating all of attributes.*

TÀI LIỆU THAM KHẢO

[1] Đỗ Văn Nhơn. Luận văn cao học: *Giải Đề trên Mạng Tính Toán*. Đại học Khoa học Tự nhiên (1996).

[2] Kiem H., Nhon Do. *Mạng tính toán và Áp dụng*. Tạp chí Tin học và điều khiển học, T.13, S.3 (1997) (10-20).

[3] Chou, S.C. & Gao, X.S. & Zhang, J.Z. *Machine Proofs in Geometry*. Singapore: Utopia Press (1994).

[4] George F. Luger & William A Stubblefield. *Artificial Intelligence*. Addison Wesley Longman, Inc. (1998).

[5] Do Van Nhon. *A Program for studying and Solving problems in Plane Geometry*.