Tạp chí Tin học và Điều khiển học, T.27, S.2 (2011), 119–130

XÂY DỰNG HỆ LÔGIC MỜ LOẠI HAI ĐẠI SỐ GIA TỬ*

PHAN ANH PHONG[1], ĐINH KHẮC ĐÔNG[2], TRẦN ĐÌNH KHANG[2]

[1]*Khoa Công nghệ thông tin, trường Đại học Vinh*
[2]*Viện Công nghệ thông tin và Truyền thông, trường Đại học Bách khoa Hà Nội*

Tóm tắt. Bài báo đề xuất một phương pháp xây dựng một hệ lôgic mờ loại hai đại số gia tử (HaT2-FLS). Quy trình này gồm hai pha, trước tiên, thiết kế một hệ lôgic mờ loại 1 (T1-FLS) từ dữ liệu bằng cách kết hợp thuật toán Fuzzy C-Means (FCM) với giải thuật di truyền (GA). Sau đó, HaT2-FLS được xây dựng từ T1-FLS vừa thiết kế. Cơ sở luật trong HaT2-FLS sử dụng cùng số luật, số tập mờ như cơ sở luật trong T1-FLS, điểm khác biệt là ở chỗ mỗi luật trong HaT2-FLS có phần tiền đề và phần kết luận đều là các HaT2FSs. Trong pha hai, các tham số của HaT2FSs được tối ưu bằng GA. Thử nghiệm phương pháp đề nghị với bài toán dự báo thời gian sống của bệnh nhân viêm tủy cho kết quả tin cậy.

Abstract. This paper proposes a method for constructing the HaT2-FLSs. This method consists of two main stages. First, the data-driven T1-FLS is designed based on combination of FCM algorithm and GA. Second, we construct HaT2-FLS on the basis of the proposed T1-FLS. Specifically, we use the rule base of T1FLS as the rule base for Ha-T2FLS and the T1-FLS's fuzzy sets as the input of HaT2-FLS. In the second stage, the parameters of HaT2FS are optimized by GA. An experiment on predicting the survival time of myeloma patients is given to show the effectiveness of the results.

1. ĐẶT VẤN ĐỀ

Tập mờ loại hai (T2FS) là sự mở rộng của tập mờ thông thường (tập mờ loại 1 - T1FS) được đưa ra đầu tiên bởi Zadeh, đó là tập mờ mà độ thuộc của mỗi phần tử cũng mờ [19]. Với ưu điểm trong việc biểu diễn và phối hợp với tính không chắc chắn nên các T2FS cũng như T2-FLS đã được ứng dụng trong nhiều bài toán khác nhau như bài toán phân lớp [6, 11], bài toán dự báo [6, 11, 13, 14],...

Theo Yager [18] "*một trong những ưu điểm của T2FS là cho phép mở rộng độ thuộc là các giá trị ngôn ngữ*" và gần đây hướng tiếp cận này đã được hưởng ứng nhiều [10, 15, 18]. Tuy nhiên, các nghiên cứu đó đều chưa đề cập đến quan hệ ngữ nghĩa giữa các độ thuộc ngôn ngữ nên kết quả thu được có thể phản trực quan [1]. Mặt khác, các nghiên cứu về T2FS có độ thuộc ngôn ngữ đều được tiếp cận theo kiểu truyền thống, đó là đặc tả các độ thuộc ngôn ngữ thành các T1FS, rồi tính toán và xử lý dựa vào các T1FS này [10, 15, 18].

Đại số gia tử (ĐSGT) là một cách phối hợp với các giá trị của biến ngôn ngữ bằng cả định

*Bài báo được hoàn thành với sự tài trợ của Quỹ Phát triển Khoa học và Công nghệ Quốc gia (Nafosted) theo mã đề tài 102.01.30.09

tính và định lượng [8, 9], một số kết quả trong [8] đã chứng tỏ tiềm năng ứng dụng của việc biểu diễn và xử lý ngôn ngữ theo hướng này. Bằng cách sử dụng các giá trị ngôn ngữ trong ĐSGT của biến chân lý ngôn ngữ TRUTH làm độ thuộc cho tập mờ ta thu được tập mờ loại hai ĐSGT (HaT2FS) [1, 2, 3]. Các kết quả thực nghiệm trong [2, 13], bước đầu đã chứng tỏ khả năng ứng dụng của dạng tập mờ này. Các phép toán tập hợp trên HaT2FS là cơ sở nền tảng của HaT2-FLS, trong các thử nghiệm [2, 13] chúng được xây dựng bằng ánh xạ định lượng ngữ nghĩa nên chưa khai thác được tính cấu trúc ĐSGT. Trong [4, 16] chúng tôi đã đề nghị một cách khác để xây dựng các phép toán tập hợp trên HaT2FS. Hướng tiếp cận này lấy quan hệ ngữ nghĩa và tính mờ của các giá trị ngôn ngữ làm cơ sở để xây dựng các phép toán. Rõ ràng, các phép toán được xây dựng theo cách này là một trong những điểm mạnh của HaT2FS, đó là, quá trình suy diễn được thao tác trực tiếp trên ngôn ngữ.

Nhằm khai thác các ưu điểm của HaT2FS, đồng thời góp phần khẳng định thêm khả năng ứng dụng của HaT2-FLS, bài báo này đề xuất một phương pháp xây dựng HaT2-FLS từ dữ liệu và thử nghiệm. Quá trình xây dựng HaT2-FLS gồm hai giai đoạn, trước hết, xây dựng một hệ lôgic mờ loại 1 (T1-FLS) dựa vào tập dữ liệu vào-ra, và sau đó, từ hệ mờ này xây dựng HaT2-FLS nhằm mục đích tăng cường sức mạnh cho hệ thống. Như vậy, công việc chính trong bài báo này là: (i) Thiết kế một T1-FLS từ dữ liệu, bao gồm: xác định số tập mờ và hàm thuộc của chúng ứng với biến vào, biến ra của hệ thống; và tạo ra một cơ sở luật. (ii) Xây dựng HaT2-FLS từ hệ T1-FLS, bao gồm: xây dựng cơ sở luật; và tối ưu các tham số của các HaT2FS. (iii) Chứng tỏ hiệu quả của HaT2-FLS trong thử nghiệm.

Sau mở đầu, Mục 2 tóm lược các kiến thức liên quan về HaT2FS. Mục 3 trình bày các thành phần của một HaT2-FLS và sau đó đề xuất một phương pháp xây dựng hệ mờ này từ dữ liệu. Thử nghiệm phương pháp đề nghị vào bài toán dự báo thời gian sống của bệnh nhân viêm tủy được trình bày trong Mục 4 và cuối cùng là phần kết luận.

2. TẬP MỜ LOẠI HAI ĐẠI SỐ GIA TỬ

2.1. Sơ lược về đại số gia tử

Theo [9], ĐSGT là bộ bốn $(AX, G, H, \leq)$, với AX là tập các giá trị, G là tập các phần tử sinh, H là tập các gia tử, bao gồm các gia tử âm (H^-) và các gia tử dương (H^+), và $\leq$ là quan hệ thứ tự giữa các phần tử của ĐSGT. Bài báo này chỉ xem xét ĐSGT đối xứng, tuyến tính, đầy đủ $(AX, G, H, \leq)$ có $G = \{c^+, c^-\}$ và tập $H = H^+ \cup H^-$ là tập sắp thứ tự, tức là $h_1 < h_2 < \ldots < h_p$ với H^+ và $h_{-1} < h_{-2} < \ldots < h_{-q}$ với H^-. Khi đó, các phần tử trong ĐSGT này đều được sắp thứ tự tuyến tính. Trong ĐSGT, mỗi $\hat{x} \in AX$ được đặc trưng bởi $fm(\hat{x})$ và $v(\hat{x})$. Giá trị $v(\hat{x}) \in [0, 1]$ phản ánh ngữ nghĩa định lượng, còn $fm(\hat{x}) \in [0, 1]$ thể hiện tính mờ của $\hat{x}$. Để biết chi tiết hơn về các đặc trưng trên có thể tham khảo trong [8], Thuật toán 2.1 sau đây được đưa ra để tính toán bộ 3 giá trị này, $\langle \underline{fm}(\hat{x}), v(\hat{x}), \overline{fm}(\hat{x}) \rangle$.

Thuật toán 2.1
Input: $fm(c^-) = \theta; \mu(h_j), j = 1, \ldots, (p+q)$; Quan hệ SIG; $\hat{x} \in AX$.
Output: $\underline{fm}(\hat{x})$, $v(\hat{x})$, $\overline{fm}(\hat{x})$.
Bước 1: Đặt $\alpha = \sum_{i=-q}^{-1} \mu(h_i)$; $\beta = \sum_{i=1}^{p} \mu(h_i)$; $\hat{x} = h_j\hat{x'}$; $h_{max} = h_p$;
$\omega(h_j\hat{x'}) = \left[1 + sign(h_j\hat{x'}) sign(h_{max}h_j\hat{x'})(\beta - \alpha)\right]/2$, với $j \neq 0$ and $-q \leq j \leq p$;
$fm(h_n h_{n-1} \ldots h_1 c) = \mu(h_n) \times \mu(h_{n-1}) \times \ldots \times \mu(h_1) \times fm(c)$;

Bước 2: Nếu $\hat{x} = c^-$ thì $v(\hat{x}) = \theta - \alpha fm(\hat{x})$; $\underline{fm}(\hat{x}) = 0$; $\overline{fm}(\hat{x}) = \theta$;
Nếu $\hat{x} = c^+$ thì $v(\hat{x}) = \theta + \alpha fm(\hat{x})$; $\underline{fm}(\hat{x}) = \theta$; $\overline{fm}(\hat{x}) = 1$;
Bước 3: Nếu $(\hat{x} \neq c^-)$ và $(\hat{x} \neq c^+)$ thì:
Nếu $(1 \leq j \leq p)$ thì $v(\hat{x}) = v(h_j\hat{x'}) = v(\hat{x'}) + sign(h_j\hat{x'})\big(\sum_{i=1}^{j} fm(h_i\hat{x'}) - \omega(h_j\hat{x'})fm(h_i\hat{x'})\big)$;
Nếu $(-q \leq j \leq -1)$ thì $v(\hat{x}) = v(h_j\hat{x'}) = v(\hat{x'}) + sign(h_j\hat{x'})\big(\sum_{i=-1}^{j} fm(h_i\hat{x'}) - \omega(h_j\hat{x'})fm(h_i\hat{x'})\big)$;
Nếu $sign(h_{max}\hat{x}) = 1$ thì $\overline{fm}(\hat{x}) = v(\hat{x}) + \beta fm(\hat{x})$;$\underline{fm}(\hat{x}) = v(\hat{x}) - \alpha fm(\hat{x})$;
Nếu $sign(h_{max}\hat{x}) = -1$ thì $\overline{fm}(\hat{x}) = v(\hat{x}) + \alpha fm(\hat{x})$; $\underline{fm}(\hat{x}) = v(\hat{x}) - \beta fm(\hat{x})$;

Sau đây là một số ký hiệu và thuật ngữ được đưa ra để sử dụng cho các phần tiếp theo:
- AX_k là tập giá trị chân lý ngôn ngữ mức k (độ dài k). Độ dài của giá trị ngôn ngữ $\hat{x}$ là số lần xuất hiện các ký hiệu trong $\hat{x}$.
- $\forall \hat{x}, \hat{y} \in AX$, $\hat{x}$ được gọi là *kế thừa ngữ nghĩa từ* $\hat{y}$, ký hiệu là $\hat{x} \lhd \hat{y}$ nếu $\hat{x}$ có dạng $\delta\hat{y}$ với δ là chuỗi gia tử bất kỳ và $\delta \neq \emptyset$.
- Hai giá trị ngôn ngữ mức-k: $\hat{x}, \hat{y} \in AX_k$ với $\hat{x} < \hat{y}$ được gọi là *cùng mức liền kề* nếu $\nexists \hat{z} \in AX_k$ sao cho $\hat{x} < \hat{z} < \hat{y}$.
- Cho m giá trị ngôn ngữ mức-k với $\hat{x}_1 < \hat{x}_2 < \ldots < \hat{x}_m$ chúng được gọi là *cùng mức liền kề nhau* nếu đôi một liên tiếp là cùng mức liền kề. Khi đó ta dùng ký hiệu $\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_m$ để biểu diễn m giá trị liền kề nhau.

2.2. Tập mờ loại hai đại số gia tử

Bằng cách sử dụng các giá trị chân lý ngôn ngữ trong ĐSGT tuyến tính, đầy đủ của biến chân lý ngôn ngữ TRUTH, $(AX, G, H, \leq)$ làm độ thuộc cho tập mờ, ta thu được tập mờ loại hai đại số gia tử (HaT2FS). Biểu diễn hình thức của tập mờ loại hai ĐSGT $\hat{A}$ trên X là: $\hat{A} = \int_X \mu_{\hat{A}}(x)/x, \mu_{\hat{A}}(x) \in AX$.

Nhận xét 2.1. Cho ĐSGT tuyến tính đầy đủ có $(p+q)$ gia tử. Nếu $\hat{x} \in AX_g$ và k nguyên và $k > g$ thì sẽ có $M = (p+q)^{k-g}$ giá trị ngôn ngữ mức-k khác nhau: $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_M$ kế thừa ngữ nghĩa từ $\hat{x}$. Gọi $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M$ là tập M phần tử lấy từ $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_M$ thỏa $\hat{x}_1 < \hat{x}_2 < \ldots < \hat{x}_M$. Khi đó $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M$ là liền kề nhau và $\hat{x} = \hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M$.

2.3. Các phép toán tập hợp trên HaT2FS

Các HaT2FS sử dụng các giá trị chân lý ngôn ngữ làm độ thuộc cho mỗi phần tử, do vậy để xây dựng phép hợp và giao trên dạng tập mờ này thì cần thiết phải đưa ra phép *join* và *meet* trên các giá trị chân lý ngôn ngữ.

Giả sử F_1 và F_2 là các tập mờ loại một trên đoạn [0,1], theo [11], phép *meet* của F_1 với F_2 được xác định bằng cách lấy mỗi phần tử trong F_1 thực hiện *meet* với mỗi phần tử trong F_2, sau đó gộp các kết quả lại. Phép *meet* này là sự mở rộng của việc áp dụng *Nguyên lý mở rộng* lên một *t*-norm, chẳng hạn như phép *min* của hai giá trị trong [0,1].

Giả sử $\hat{x} \in AX_g$ và k là một số nguyên thỏa $k > g$, theo Nhận xét 2.1, $\hat{x}$ có thể được xem là một tập hợp gồm M giá trị ngôn ngữ mức-k kế thừa ngữ nghĩa từ nó, $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M$. Tương tự cách xây dựng phép *meet* trên các T1FS, *meet* của $\hat{x}$ và $\hat{y} \in AX_k$ sẽ được tính như sau. Trước tiên, lấy mỗi phần tử $\hat{x}_i$ trong M giá trị ngôn ngữ mức-k kế thừa ngữ nghĩa từ $\hat{x}$ thực hiện *meet* với $\hat{y}$, sau đó gộp các kết quả lại. Ở đây, phép *meet* giữa $\hat{x}_i$ và $\hat{y}$ được thực hiện bằng cách áp dụng *Nguyên lý mở rộng* của phép *min*.

Lập luận tương tự phép *meet*, phép *join* các giá trị chân lý ngôn ngữ trên AX sẽ được xây dựng, ở đây áp dụng *Nguyên lý mở rộng* cho phép *max* thay vì phép *min*.

Giả sử $\hat{x} \in AX_g$, $\hat{y} \in AX_k$, và $k \geq g$, khi đó có hai khả năng xảy ra: (i) $\hat{y}$ kế thừa ngữ nghĩa từ $\hat{x}$, tức là tồn tại $\hat{x}_i$ trong M giá trị ngôn ngữ mức-k kế thừa ngữ nghĩa từ $\hat{x}$ đúng bằng $\hat{y}$, và (ii) $\hat{y}$ và $\hat{x}$ không có quan hệ kế thừa ngữ nghĩa, tức là $y < \hat{x}_1$ hoặc là $y > \hat{x}_M$.

Định nghĩa 2.1. Giả sử $\hat{x} \in AX_g, \hat{y} \in AX_k$ và $k \geq g$ và $\hat{x}$ được biểu diễn bằng M giá trị ngôn ngữ mức-k: $\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M$. Khi đó, phép *meet* ($\vartriangle$) giữa $\hat{x}$ và $\hat{y}$ được định nghĩa:
(i) $\hat{x} \vartriangle \hat{y} = (\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M) \vartriangle \hat{y} = \hat{y}$ nếu $\hat{y} < \hat{x}_1$.
(ii) $\hat{x} \vartriangle \hat{y} = (\hat{x}_1 \vartriangle \hat{y})\Theta(\hat{x}_2 \vartriangle \hat{y})\ldots\Theta(\hat{x}_M \vartriangle \hat{y}) = \hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_i$ nếu $\hat{y} = \hat{x}_i$ với $i \in [1, M]$.
(iii) $\hat{x} \vartriangle \hat{y} = (\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M) \vartriangle \hat{y} = \hat{x}$ nếu $\hat{y} > \hat{x}_M$.

Định nghĩa 2.2. Giả sử $\hat{x} \in AX_g, \hat{y} \in AX_k$ và $k \geq g$ và $\hat{x}$ được biểu diễn bằng M giá trị ngôn ngữ mức-k: $\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M$. Khi đó, phép *join* ($\triangledown$) giữa $\hat{x}$ và $\hat{y}$ được định nghĩa:
(i) $\hat{x}\triangledown\hat{y} = (\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M)\triangledown\hat{y} = \hat{x}$ nếu $\hat{y} < \hat{x}_1$.
(ii) $\hat{x}\triangledown\hat{y} = (\hat{x}_1\triangledown\hat{y})\Theta(\hat{x}_2\triangledown\hat{y})\ldots\Theta(\hat{x}_M\triangledown\hat{y}) = \hat{x}_i\Theta\hat{x}_{i+1}\Theta\ldots\Theta\hat{x}_M$ nếu $\hat{y} = \hat{x}_i$ với $i \in [1, M]$.
(iii) $\hat{x}\triangledown\hat{y} = (\hat{x}_1\Theta\hat{x}_2\Theta\ldots\Theta\hat{x}_M)\triangledown\hat{y} = \hat{y}$ nếu $\hat{y} > \hat{x}_M$.

Kết quả thu được của các phép toán trên không phải lúc nào cũng là một phần tử của ĐSGT. Cụ thể hơn, khi $\hat{x}$ và $\hat{y}$ có quan hệ kế thừa ngữ nghĩa, như trong trường hợp (ii), thì gộp của các phần tử từ $\hat{x}_1$ đến $\hat{x}_i$ (với phép *meet*) hoặc gộp của các phần tử từ $\hat{x}_i$ đến $\hat{x}_M$ (với phép *join*) chưa chắc đã là một phần tử của ĐSGT. Để kết quả của phép hợp và giao các HaT2FS bất kỳ cũng là một HaT2FS, ta phải xây dựng cách tính xấp xỉ phép *join* và *meet* của các giá trị chân lý ngôn ngữ và vấn đề này đã được bàn luận chi tiết trong [4]. Để dễ theo dõi, ta ký hiệu $\tilde{\vartriangle}$ và $\tilde{\triangledown}$ là phép tính gần đúng *meet* và *join* của hai giá trị ngôn ngữ tương ứng.

Thuật toán 2.2. Tính *meet* của hai giá trị chân lý ngôn ngữ
Input: $H^- = \{h_{-q}, \ldots, h_{-1}\}; H^+ = \{h_1, \ldots, h_p\}$; Bảng *SIG*; $\hat{x} \in AX_g, \hat{y} \in AX_k, k \geq g$;
Output: $\hat{z} = \hat{x}\tilde{\vartriangle}\hat{y}$
Bước 1: Nếu $\hat{y}$ không kế thừa ngữ nghĩa từ $\hat{x}$ thì :
- Nếu $(\hat{y} < \hat{x})$ thì *Return* $\hat{z} = \hat{y}$, ngược lại, *Return* $\hat{z} = \hat{x}$;
Ngược lại: Giả sử $\hat{x} = \delta c$ và $\hat{y} = h_n h_{n-1}\ldots h_1\hat{x}$ với δ là chuỗi gia tử bất kỳ;
- Ký hiệu $\hat{x}_k^{min}$ là giá trị ngôn ngữ mức-k nhỏ nhất kế thừa ngữ nghĩa từ $\hat{x}$; Đặt $j = 0$;
Bước 2: Nếu $(j = n)$ thì chuyển sang Bước 3, ngược lại $(j < n)$ thì :
- Xác định $\hat{x}_{g+j}^{min}$; Nếu $\hat{y} \geq \hat{x}_{g+j}^{min}$ thì *Return* $\hat{z} = \hat{x}_{g+j}^{min}$; Ngược lại, $j = j + 1$; Lặp lại Bước 2 ;
Bước 3: *Return* $\hat{z} = \hat{y}$.

Thuật toán 2.3. Tính *join* của hai giá trị chân lý ngôn ngữ
Input: $H^- = \{h_{-q}, \ldots, h_{-1}\}; H^+ = \{h_1, \ldots, h_p\}$; Bảng *SIG*; $\hat{x} \in AX_g, \hat{y} \in AX_k, k \geq g$;
Output: $\hat{z} = \hat{x}\tilde{\triangledown}\hat{y}$
Bước 1: Nếu $\hat{y}$ không kế thừa ngữ nghĩa từ $\hat{x}$ thì :
- Nếu $(\hat{y} < \hat{x})$ thì *Return* $\hat{z} = \hat{x}$; ngược lại, *Return* $\hat{z} = \hat{y}$;
Ngược lại: Giả sử $\hat{x} = \delta c$ và $\hat{y} = h_n h_{n-1}\ldots h_1\hat{x}$ với δ là chuỗi gia tử bất kỳ;
- Ký hiệu $\hat{x}_k^{max}$ là giá trị ngôn ngữ mức-k lớn nhất kế thừa ngữ nghĩa từ $\hat{x}$; Đặt $j = 0$;
Bước 2: Nếu $(j = n)$ thì chuyển sang Bước 3, ngược lại $(j < n)$ thì :
- Xác định $\hat{x}_{g+j}^{max}$; Nếu $\hat{y} \leq \hat{x}_{g+j}^{max}$ thì *Return* $\hat{z} = \hat{x}_{g+j}^{max}$; Ngược lại, $j = j + 1$; Lặp lại Bước 2 ;
Bước 3: *Return* $\hat{z} = \hat{y}$..

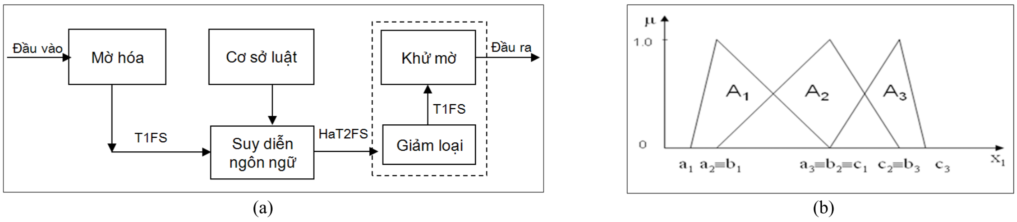
Giả sử $\hat{A} = \sum_{i=1}^{N} \mu_{\hat{A}}(x_i)/x_i$ và $\hat{B} = \sum_{i=1}^{N} \mu_{\hat{B}}(x_i)/x_i$ là các HaT2FS trên X. Khi đó:
(i) Phép hợp: $\hat{A} \cup \hat{B} = \sum_{i=1}^{N} \left(\mu_{\hat{A}}(x_i)\tilde{\triangledown}\mu_{\hat{B}}(x_i)\right)/x_i$

(ii) Phép giao: $\hat{A} \cap \hat{B} = \sum_{i=1}^{N} \big(\mu_{\hat{A}}(x_i)\tilde{\Delta}\mu_{\hat{B}}(x_i)\big)/x_i$
(iii) Phần bù của $\hat{A}$: $\neg\hat{A} = \sum_{i=1}^{N} \overline{\mu_{\hat{A}}(x_i)}/x_i$, trong đó, nếu $\mu_{\hat{A}}(x_i) = \delta c$ thì $\overline{\mu_{\hat{A}}(x_i)} = \delta\overline{c}$.

3. XÂY DỰNG HỆ LÔGIC MỜ LOẠI HAI ĐẠI SỐ GIA TỬ

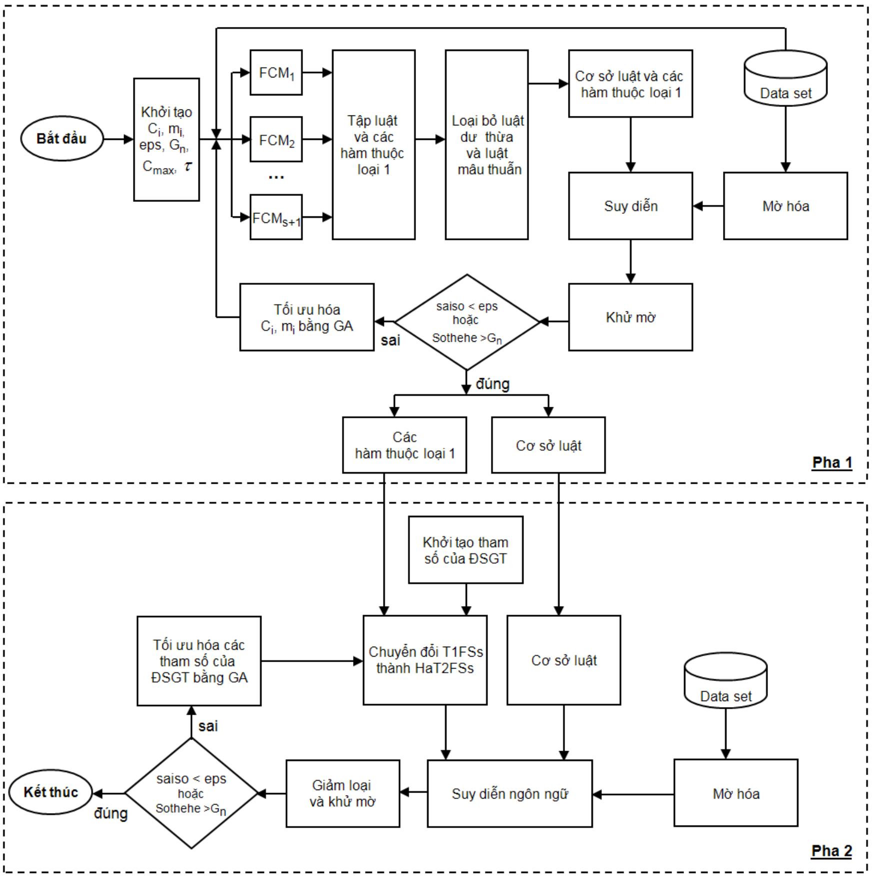
3.1. Hệ lôgic mờ loại hai đại số gia tử

Tương tự các hệ lôgic mờ loại hai dựa trên luật [11], các thành phần của HaT2-FLS được minh hoạ trong Hình 3.1 (a), gồm 4 khối: mờ hóa; cơ sở luật; mô tơ suy diễn và khối xử lý đầu ra (giảm loại và khử mờ).



Hình 3.1. (a) Hệ lôgic mờ loại hai đại số gia tử. (b) Tập tham số của hàm thuộc loại 1

Khối *mờ hóa* nhận đầu vào là các giá trị rõ và đầu ra là một tập mờ loại hai ĐSGT tương ứng biểu diễn giá trị đó khi cho trước một ĐSGT của biến chân lý TRUTH.
Cơ sở luật chứa các luật *NẾU-THÌ*, giả sử mỗi luật có s giả thiết và một kết luận, và luật thứ l có dạng như sau: *NẾU* x_1 *là* $\hat{F}_1^l$ *và* x_2 *là* $\hat{F}_2^l$ ... x_s *là* $\hat{F}_s^l$ *THÌ* y *là* $\hat{G}^l$, trong đó, $x_1, x_2, \ldots, x_s$ là các biến vào lần lượt xác định trên không gian $U_1, U_2, \ldots, U_s$; y là biến ra xác định trên không gian V. $\hat{F}_1^l, \ldots, \hat{F}_s^l$ là các HaT2FS xác định trên các không gian tương ứng.
Để hiểu rõ quy trình suy diễn trong HaT2-FLS, ta xét cơ sở luật gồm M luật sau đây:
Luật thứ l: NẾU x là $\hat{A}^l$ THÌ y là $\hat{B}^l$
Sự kiện: x là $\hat{A}_0$
Kết luận: y là $\hat{B}_0$
ở đây, x là biến đầu vào, y là biến đầu ra; $\hat{A}^l, \hat{B}^l, \hat{A}_0, \hat{B}_0$ là các HaT2FS.
Tương tự như suy diễn mờ, từ luật *NẾU-THÌ* thứ l hình thành một quan hệ mờ $\hat{R}^l$, quan hệ này cũng là một HaT2FS. Giả sử có quan hệ $\hat{R}_C^l = \big(\hat{A}^l \times \hat{B}^l\big) = \int_{U\times V} \big(\mu_{\hat{A}^l}(u)\tilde{\Delta}\mu_{\hat{B}^l}(v)\big)/(u,v)$. Khi đó, đầu ra của luật thứ l, $\hat{B}_0^l$, ứng với đầu vào $\hat{A}_0$ được tính bằng phép hợp thành giữa tập mờ $\hat{A}_0$ và quan hệ mờ $\hat{R}_C^l$: $\hat{B}_0^l = \hat{A}_0 \circ \hat{R}_C^l$, $\mu_{\hat{B}_0^l}(v) = \tilde{\nabla}_u\big(\mu_{\hat{A}_0}(u)\tilde{\Delta}\mu_{\hat{R}_C^l}(u,v)\big)$. Cuối cùng, để xác định tập mờ đầu ra của hệ mờ $\hat{B}_0$ ứng với tập mờ đầu vào $\hat{A}_0$, tức là đốt cháy, $\hat{A}_0$ bởi tất cả các luật có trong cơ sở luật, ta sử dụng phép hợp các HaT2FS đầu ra của mỗi luật, $\mu_{\hat{B}_0^l}(v)$, tức là: $\hat{B}_0 = \tilde{\nabla}_{l=1}^{M}\big[\mu_{\hat{B}_0^l}(v)\big]$.
Bộ phận xử lý đầu ra gồm hai phần: *giảm loại* và *khử mờ*. Phép giảm loại HaT2FS thành T1FS nhờ vào ánh xạ định lượng ngữ nghĩa v trong [8], còn quá trình giải mờ được thực hiện giống với T1FS.



Hình 3.2. Sơ đồ xây dựng hệ lôgic mờ loại hai đại số gia tử

3.2. Xây dựng hệ lôgic mờ loại hai đại số gia tử

Bài báo này tiếp cận theo từng phần để xây dựng một Ha-T2FLS từ dữ liệu. Trước tiên, kết hợp thuật toán FCM với GA thiết kế một T1-FLS và sau đó nhờ vào hệ mờ này để xây dựng HaT2-FLS. Quy trình để xây dựng hệ lôgic mờ này được minh họa trong Hình 4.1.

Thuật toán FCM là một trong những thuật toán phân cụm được đưa ra bởi Bezdek [5]. Quá trình phân cụm có thể đạt được tối ưu dựa trên việc cực tiểu hóa hàm $J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m \parallel x_i - c_j \parallel$, trong đó, c là số cụm, u_{ij} thể hiện độ thuộc của x_i trong cụm j , x_i là mẫu thứ i trong N mẫu dữ liệu, c_j là tâm của cụm thứ j, $m > 1$ được gọi là chỉ số tính mờ và $\parallel \star \parallel$ là một hàm đo khoảng cách giữa các mẫu dữ liệu với tâm cụm.

GA là giải thuật hoạt động tương tự cơ chế chọn lọc và di truyền tự nhiên [7]. Khi sử dụng GA để giải bài toán tối ưu, phụ thuộc vào bộ 4 tham số (S, p_c, p_m, G_n), trong đó S – số cá thể trong quần thể; p_c – xác suất lai ghép, p_m – xác suất đột biến và G_n – số thế hệ cần tiến hóa. Cá thể có giá trị hàm mục tiêu tốt nhất của mọi thế hệ sẽ là lời giải cuối cùng của GA. Trong bài báo này, GA được dùng để tối ưu các tham số c và m trong thuật toán FCM ở Pha 1, và các tham số của HaT2FS ở Pha 2.

3.2.1. Pha 1 - Xây dựng hệ lôgic mờ loại 1 từ dữ liệu

Để đơn giản, ta xét mô hình mờ MISO (Multi-Input Single-Ouput) với 2 biến vào x_1, x_2 và 1 biến ra y. Giả sử k_1, k_2 và k_3 là số cụm tương ứng thu được khi sử dụng thuật toán FCM để phân cụm tập dữ liệu của các biến x_1, x_2 và y. Khi đó, các biến x_1, x_2 và y sẽ có k_1, k_2 và k_3 tập mờ tương ứng. Hơn nữa, nhờ vào tâm mỗi cụm và sự phân bố dữ liệu trong các cụm ta còn xác định được các tham số của các tập mờ này, khi dạng hàm thuộc của tập mờ cho trước. Chẳng hạn, xét biến x_1, giả sử $k_1 = 3$ và hàm thuộc có dạng tam giác, khi đó 3 tập mờ loại một A_1, A_2 và A_3 được mô tả theo bộ 3 tham số (a_i, b_i, c_i), với $i = 1, 2, 3$. Không mất tính tổng quát, giả sử A_2 nằm giữa A_1 và A_3, A_1 nằm bên trái và A_3 nằm bên phải A_2 (xem Hình 3.1.b). Từ đây, ta có thể đề xuất một cách để xác định các tham số cho 3 tập mờ này. Chẳng hạn, tại các điểm b_1, b_2 và b_3 chúng đều có độ thuộc là 1 tại vị trí trùng với tâm của mỗi cụm; $a_1 = min_{x1} - 30\% \times (max_{x1} - min_{x1}); c_1$ trùng với $b_2; a_2$ trùng với $b_1; c_2$ trùng với $b_3; a_3$ trùng với b_2; còn $c_3 = max_{x1} + 30\% \times (max_{x1} - min_{x1})$. Trong đó, min_{x1} là giá trị nhỏ nhất và max_{x1} là giá trị lớn nhất của tập dữ liệu ứng với biến x_1.

Như vậy, từ kết quả của quá trình phân cụm, một tập luật gồm $k_1 \times k_2 \times k_3$ luật được tạo ra. Tuy nhiên tập luật này còn mâu thuẫn. Để tạo ra một tập luật "tốt hơn" tập luật trên, ta sử dụng bộ dữ liệu huấn luyện, cụ thể như sau.

Đọc bản ghi thứ t, (v_{1t}, v_{2t}, v_t) từ bộ dữ liệu. Đốt cháy chúng qua từng luật, gán trọng số w_j cho mỗi luật. Giả sử luật j: *NẾU* x_1 *là* A_1 *và* x_2 *là* A_2 *THÌ* y *là* B, khi đó, trọng số của luật này tương ứng với bản ghi t là: $w_{tj} = min\{\mu_{A_1}(v_{1t}), \mu_{A_2}(v_{2t}), \mu_B(v_t)\}$. Như vậy, trọng số của luật j khi đốt cháy N bản ghi: $w_j = \frac{1}{N}\sum_{j=1}^{N} w_{tj}$. Đối với các luật mâu thuẫn (luật có phần tiền đề giống nhau nhưng phần kết luận khác nhau) thì giữ lại luật có trọng số lớn nhất. Tuy nhiên, sau khi loại bỏ các luật mâu thuẫn, $k_1 \times k_2$ luật còn lại vẫn có thể còn dư thừa. Lúc này, một ngưỡng $\tau \in [0, 1]$ được chọn để tiếp tục loại bỏ các luật có trọng số quá nhỏ, ngưỡng này tùy thuộc vào bài toán cụ thể.

Như đã biết, khi thuật toán FCM thực hiện thì hai tham số c và m cần phải được xác định. Nhiều ứng dụng sử dụng thuật toán FCM chọn $m = 2$ theo gợi ý của Bezdek từ năm 1976 [5]. Gần đây trong [12] Pal và Bezdek đã chỉ ra rằng m có thể được lựa chọn trong [1.5, 2.5], hay trong [17] khuyến cáo m nên lấy trong đoạn [1.5, 4],... Như vậy, việc chọn giá trị cho tham số m phụ thuộc vào đặc điểm của bộ dữ liệu. Ở đây, ta sử dụng GA để tối ưu hai tham số m và c sao cho sai số của hệ mờ loại một được xây dựng từ kết quả phân cụm là nhỏ nhất. Để tránh bùng nổ tập luật, ở đây tham số c được giới hạn trong khoảng [2, c_{max}]. Gọi **CMOptimization**(.) là thủ tục tối ưu tham số c_i và m_i của thuật toán FCM. Lúc này, hàm thích nghi được chọn là LSE (Least Square Error): $LSE = \sqrt{\sum_{t=1}^{N}(y_t - v_t)^2}$, trong đó, N – số bản ghi huấn luyện, y_t - giá trị đầu ra của hệ mờ loại 1, v_t – giá trị đầu ra mong muốn trong bản ghi thứ t.

Lưu ý rằng, quá trình tính toán trong T1-FLS sử dụng mờ hóa đơn trị, toán tử t-norm min, t-conorm max và khử mờ trọng tâm.

Theo cách mô tả trên đây, Thủ tục 3.1 được đưa ra để thiết kế một T1-FLS từ dữ liệu.

Thủ tục 3.1. Thiết kế hệ mờ loại 1

Input: N bản ghi, mỗi bản ghi có $s + 1$ thành phần, với s các biến vào $x_1, \ldots, x_s$; và thành phần còn lại là biến ra y; G_n - Số thế hệ; ε.

Output: Một cơ sở luật và các hàm thuộc loại 1

Bước 1: Với $i = 1, \ldots, s+1$ khởi tạo $c_i = 2$; $m_i = 2$; ϵ; τ; $G_n = 100$; $C_{max} = \lfloor \sqrt{N} \rfloor$
Bước 2: Xác định các hàm thuộc của các tập mờ loại 1
- Phân cụm bằng thuật toán FCM độc lập trên dữ liệu của từng biến vào và biến ra theo (c_1, m_1), (c_2, m_2), ..., (c_{s+1}, m_{s+1}). Giả sử thu được $k_1, k_2, \ldots, k_{s+1}$ cụm tương ứng.
- Xác định các hàm thuộc tam giác của các tập mờ loại một A_k và B_k từ kết quả phân cụm
Bước 3: Xác định tập luật: Từ các tập mờ A_k và B_k ta có $k_1 \times k_2 \times \ldots \times k_{s+1}$ luật.
Giả sử luật thứ i có dạng: *NẾU* x_1 *là* A_1^j *và* $\ldots$ *và* x_s *là* A_s^j *THÌ* y *là* B^j
Bước 4: Loại bỏ luật mâu thuẫn và luật dư thừa để có một cơ sở luật tối ưu
Bước 5: Xác định sai số đầu ra của hệ mờ loại 1
- *Tổng_sai_số* $= 0$; $t = 1$;
- Trong khi $t \leq N$: Đọc bản ghi thứ t, $(v_{1t}, v_{2t}, \ldots, v_{st}, v_t)$;
+ Tính đầu ra rõ y_t theo hệ lôgic mờ ở Bước 4 với đầu vào là : $(v_{1t}, v_{2t}, \ldots, v_{st})$.
+ *Tổng_sai_số* = *Tổng_sai_số* + $(y_t - v_t)^2$; $t = t + 1$;
- Tính toán sai số của hệ: *Sai_số* = (*Tổng_sai_số*)$^{1/2}$
Bước 6: Tối ưu c_i, m_i bằng GA
Nếu (*Sai_số* $< \varepsilon$) hoặc (*Số_thế_hệ* $> G_n$) thì chuyển sang Bước 7;
Ngược lại, **CMOptimization**(c_i, m_i); Quay lên Bước 2 ;
Bước 7: Một hệ lôgic mờ loại 1.

3.2.2. Pha 2 - Xây dựng Ha-T2FLS

Mờ hóa: Thực chất của bước mờ hóa trong pha này là chuyển tập mờ loại một thành tập mờ loại hai ĐSGT, trong đó cần chuyển các độ thuộc, là các giá trị trong khoảng [0,1] thành các giá trị chân lý ngôn ngữ của ĐSGT. Theo cách tiếp cận của ĐSGT, độ dài của giá trị ngôn ngữ, k, thể hiện tính mờ của nó. Hay nói cách khác k phản ánh mức độ không chắc chắn trong giá trị ngôn ngữ, do vậy, trong bước này chúng tôi chọn độ dài lớn nhất của một giá trị chân lý ngôn ngữ là k_{max} với mục đích duy trì một mức độ không chắc chắn.

Để thực hiện được việc mờ hóa trong pha này, trước tiên ta xác định tập giá trị ngôn ngữ mức$-k$ với $k = 1, 2, \ldots, k_{max}$, sau đó tìm giá trị ngôn ngữ mà khoảng tính mờ của nó bao hàm độ thuộc này đồng thời định lượng ngữ nghĩa của giá trị ngôn ngữ gần độ thuộc đó nhất. Chi tiết của bước này được trình bày theo Thủ tục 3.2 sau đây.

Thủ tục 3.2. Chuyển đổi T1FS thành HaT2FS
Input: Tập tham số tính mờ của ĐSGT: $fm(c^-), \mu(h_j)$ với $j = 1, \ldots, p+q$, trong đó ($p+q$ là số các gia tử); k_{max} nguyên dương; tập mờ loại một A xác định trên X.
Output: Tập mờ loại hai ĐSGT: $\hat{A}$
Với mỗi x_i thuộc vào *giá đỡ* của tập mờ A, thực hiện các bước sau:
Bước 1: $k = 1; temp = \emptyset; temp$ là các giá trị chân lý ngôn ngữ kết quả tiềm năng.
Bước 2: Xác định M giá trị chân lý ngôn ngữ mức-k : $\hat{x}_{1k}, \ldots, \hat{x}_{Mk}$, với $M = 2 \times (p+q)^{k-1}$
Bước 3: Xây dựng M khoảng $\big(\underline{fm}(\hat{x}_{1k}), \overline{fm}(\hat{x}_{1k})\big]; \ldots; \big(\underline{fm}(\hat{x}_{Mk}), \overline{fm}(\hat{x}_{Mk})\big]$ tương ứng với $\hat{x}_{1k}, \ldots, \hat{x}_{Mk}$ theo Thuật toán 2.1
Bước 4: Nếu $\underline{fm}(\hat{x}_{ik}) < \mu_A(x_i) \leq \overline{fm}(\hat{x}_{ik})$ thì $temp = temp \cup \{\hat{x}_i\}$
Bước 5: Nếu $k < k_{max}$ thì $k = k + 1$ và quay lên Bước 2
Bước 6: Xác định giá trị chân lý ngôn ngữ $\hat{x}_j \in temp$ sao cho $|\mu_A(x_i) - v(\hat{x}_j)|$ nhỏ nhất.

Cơ sở luật và mô tơ suy diễn: Cơ sở luật trong HaT2-FLS được tạo ra từ cơ sở luật của hệ mờ ở Pha 1, nó sử dụng cùng số luật và cùng số tập mờ như T1-FLS, điểm khác biệt là ở

chỗ các tập mờ trong phần tiền đề và phần kết luận của mỗi luật đều là các HaT2FS. Thủ tục 3.2 sẽ đảm nhận việc chuyển đổi các tập mờ loại một thành các HaT2FS, còn quá trình suy diễn được thực hiện nhờ vào các phép toán tập hợp trên HaT2FS.
Xử lý đầu ra: Sử dụng hàm định lượng ngữ nghĩa v trong ĐSGT để giảm loại HaT2FS thành T1FS, sau đó áp dụng phép khử mờ trọng tâm ta thu được đầu ra y của hệ.
Tối ưu các tham số của HaT2FS: Các tham số của HaT2FS được tối ưu bằng giải thuật di truyền. Ký hiệu thủ tục này là **HaParaOptimization**$(fm(c^-), \mu(h_j))$ với $j = 1, \ldots, p+q$. Mã hóa lời giải thành chuỗi bits, mỗi tham số trong thủ tục này dùng 10 bits. Như vậy, độ dài của mỗi nhiễm sắc thể là $10 \times (p+q+1)$ bits. Để đảm bảo ràng buộc $\sum_{j=1}^{p+q} \mu(h_j) = 1$, mỗi $\mu(h_j)$ sẽ được mã hóa, sau đó thêm bước giải mã chúng bằng cách chia tỷ lệ. Mỗi đoạn 10 bits của từng $\mu(h_j)$ ký hiệu là hj_2 sẽ được chuyển sang hệ thập phân là $decimal(hj_2)$. Ký hiệu tổng thập phân của các $\mu(h_j)$ này là $decimal(sum)$. Khi đó, giá trị thực để tính toán của mỗi $\mu(h_j)$ được tính theo công thức: $\mu(h_j) = decimal(hj_2)/decimal(sum)$.
Hàm thích nghi trong thủ tục tối ưu **HaParaOptimization**$(fm(c^-), \mu(h_j))$ được sử dụng cũng là hàm LSE: $LSE = \sqrt{\sum_{t=1}^{N}(Y_t' - Y_t)^2}$, ở đây, Y_t' là đầu ra rõ của Ha-T2FLS ứng với bản ghi thứ t và Y_t là đầu ra mong muốn.
Như vậy, toàn bộ quá trình xây dựng hệ lôgic mờ loại hai đại số gia tử có thể được hình thức hóa theo thủ tục sau đây.

Thủ tục 3.3. Xây dựng Ha-T2FLS
Input: $fm(c^-), \mu(h_j)$ với j thỏa: $-q \le j \le p, j \ne 0$; $\epsilon; \tau; k_{max}; G_n$-Số thế hệ cần tiến hóa.
Output: Một hệ logic mờ loại hai ĐSGT
Bước 1: Sử dụng Thủ tục 3.1 để thiết kế một hệ lôgic mờ loại một từ tập dữ liệu
Bước 2: Khởi tạo Ha-T2FLS
2.1. Chuyển các T1FSs trong Bước 1 thành các HaT2FSs theo Thủ tục 3.2.
2.2. Kết hợp cơ sở luật trong Bước 1 với các HaT2FS trong Bước 2.1 ta có một Ha-T2FLS
Bước 3: Tính toán sai số của Ha-T2FLS
3.1. *Tổng_sai_số* $= 0$; $t = 1$;
3.2. Trong khi $t \le N$: Đọc bản ghi thứ t, $(v_{1t}, v_{2t}, \ldots, v_{st}, v_t)$;
- Tính đầu ra rõ Y_t' theo Ha-T2FLS ở Bước 2.2 với đầu vào: $(v_{1t}, v_{2t}, \ldots, v_{st})$:
  + Suy diễn, giảm loại HaT2FS bằng hàm v, và khử mờ trọng tâm T1FS ta có Y_t';
  + *Tổng_sai_số* = *Tổng_sai_số* + $(Y_t' - v_t)^2$; $t = t + 1$;
3.3. Tính toán sai số của hệ: *Sai_số* = $($*Tổng_sai_số*$)^{1/2}$
Bước 4: Tối ưu tham số của Ha-T2FLS
Nếu (*Sai_số* $< \varepsilon$) hoặc (*Số_thế_hệ* $> G_n$) thì chuyển đến Bước 5
Ngược lại, **HaParaOptimization**$(fm(c^-), \mu(h_j))$; Chuyển lên Bước 2;
Bước 5: Hệ lôgic mờ loại hai đại số gia tử tối ưu.

4. THỬ NGHIỆM VÀ BÀN LUẬN

4.1. Bài toán

Bộ dữ liệu về bệnh nhân viêm tủy có thể được tìm thấy trong tài liệu SAS/STAT 9.2 [20]. Theo bộ dữ liệu MYELOMA, biến Time mô tả thời gian sống tính bằng tháng của bệnh nhân. Biến Vstatus nhận hai giá trị 0 và 1, thể hiện bệnh nhân còn sống hay đã chết sau quá trình nghiên cứu. Nếu Vstatus nhận giá trị 0 thì Time tương ứng là thiếu (vì bệnh nhân còn sống).

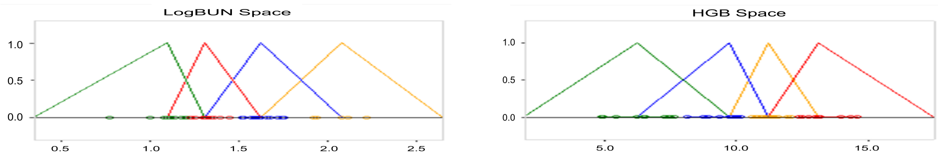
Bảng 1. LSE của các hệ lôgic mờ

Fuzzy Systems	T1-FLS [14]	IT2-FLS [14]	SIT2-FLS [14]	Ha-T2FLS
LSE	2.28	2.04	1.78	1.75

Các biến liên quan đến sự sống của bệnh nhân là LogBUN – giá trị logarithm của urát nitơ trong máu, HGB – Hemoglobin, Platelet – tiểu cầu khi phân tích, Age – tuổi của bệnh nhân, LogWBC – log của WBC khi phân tích, Frac – các vết rạn xương, LogPBM – log phần trăm của tế bào huyết tương trong tủy xương, Protein – hiện tượng nước tiểu có protein và Scalc – huyết thanh canxi.

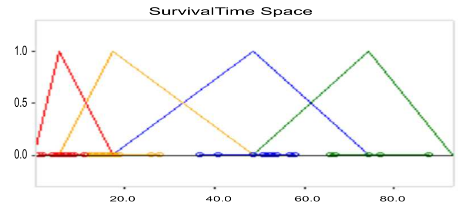
4.2. Thử nghiệm và bàn luận

Mặc dù các tham số trong bài toán trên đều liên quan đến thời gian sống của bệnh nhân nhưng trong các nghiên cứu, hai nhân tố có ảnh hưởng lớn nhất là LogBUN và HGB. Để tiện cho việc so sánh, thử nghiệm này lựa chọn tham số và bộ dữ liệu hoàn toàn tương tự [14]. Trong số 65 bộ dữ liệu này, 45 bộ được dùng để xây dựng mô hình, 20 bộ còn lại được dùng để kiểm tra. Trong Pha 1, điều kiện dừng của FCM là $eps = 10^{-3}$. Thủ tục **CMOptimization**(.) tối

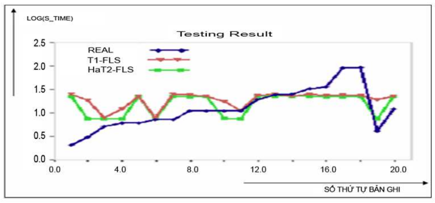


Hình 4.1. Hàm thuộc của T1FS ứng với các biến LogBUN, HGB

ưu c_i, m_i , với $i = 1, 2, 3$, tương ứng với 2 biến vào LogBUN và HGB và 1 biến ra S_TIME. Mỗi c_i được mã hóa thành 3 bits (vì $c_i \in [2, 7]$), mỗi m_i được mã hóa thành 10 bits (vì $m_i \in [1.10, 10.00]$). Áp dụng lai ghép đơn điểm $p_c = 95\%$. Xác suất đột biến $p_m = 1\%$. Số thế hệ $G_n = 100$. Tại mỗi thế hệ giữ lại 10 cá thể tốt nhất cho thế hệ sau. Hàm thích nghi là LSE. Sau khi phân cụm mờ tối ưu tập dữ liệu của các biến vào LogBUN, HGB và biến ra S_TIME, mỗi tập dữ liệu đều được phân thành 4 cụm ($c_1 = c_2 = c_3 = 4$), tương ứng với các tham số $m_1 = 2.67$; $m_2 = 2.98$ và $m_3 = 1.71$. Khi đó, các tập mờ ứng với 2 biến vào và 1 biến ra được biểu diễn theo dạng tam giác (Hình 4.1). Lúc này, cơ sở luật của T1-FLS gồm 16 luật và sai số huấn luyện LSE = 3.25, còn sai số kiểm tra LSE = 2.20. Từ hệ mờ loại một này, HaT2-FLS được xây dựng. Trong Pha 2, ta xét ĐSGT $AT = (AX, G, H, \leq)$ với các tham số $H^+ = \{more, very\}$; $H^- = \{less, possibly\}$ và các giá trị khởi tạo $fm(c^-) = 0.5$, $\mu(less) = \mu(possibly) = \mu(more) = \mu(mery) = 0.25$. Các tham số $p_c, p_m, G_n, \ldots$ và hàm thích nghi trong thủ tục tối ưu **HaParaOptimization**($fm(c^-), \mu(h_j)$) được sử dụng giống như trong thủ tục **CMOptimization**(.). Ngoài ra, nó có một số điểm khác. Thứ nhất, cách mã hóa vẫn là nhị phân nhưng mỗi nhiễm sắc thể trong thủ tục này là 50 (vì ĐSGT đang xét là đối xứng và có 4 gia tử *less*, *possibly*, *more* và *very*). Thứ hai, để giảm thời gian chạy của chương trình, tham số $fm(c^-)$ được giới hạn trong khoảng [0.3, 0.7] và cuối cùng là, bổ sung bước giải mã theo tỷ lệ của các $\mu(h_j)$ để đảm bảo ràng buộc $\sum_{j=1}^{4} \mu(h_j) = 1$. Khi các tham số của ĐSGT, $fm(c^-) = 0.5$, $\mu(less) = 0.39$, $\mu(possibly) = 0.23$, $\mu(more) = 0.33$ và $\mu(very) = 0.05$ thì hệ mờ loại hai ĐSGT cho kết quả tốt, với sai số huấn luyện LSE = 3.12 và sai số kiểm tra LSE = 1.75. So sánh với hệ mờ loại hai khoảng thống kê (SIT2-FLS) trong [14] thì sai số của Ha-T2FLS có tốt hơn chút ít (của [14], LSE = 1.78), tuy nhiên so sánh với sai số của hệ mờ loại 1 và hệ mờ loại hai khoảng (IT2-FLS) mà nhóm tác giả trong [14] đã công bố thì kết quả dự báo của Ha-T2FLS là thực sự khả quan (Bảng 1).



(a)



(b)

Hình 4.2.a *Hình 4.2.b*

Hình 4.2.a. Hàm thuộc của T1FS ứng với biến S-Time.

Hình 4.2.b. Dự báo thời gian sống của bệnh nhân viêm tủy với các hệ mờ khác nhau.

Hình 4.2.b cho thấy rõ hơn về kết quả dự đoán của 20 bản ghi với 2 hệ lôgic mờ khác nhau: T1-FLS, Ha-T2FLS và kết quả thực tế. Trong 20 bản ghi được dùng để kiểm tra, hơn một nửa có kết quả dự báo gần với thực tế. Kết quả dự báo của các bản ghi 1, 2, 5, 7, 17, 18 có sai lệch đáng kể so với thực tế, đây cũng là nhược điểm tương tự kết quả dự báo trong [14], việc lựa chọn hai tham số LOGBUN và HGB có thể chưa thể hiện tốt cho bài toán dự báo này, hoặc là dữ liệu của các bản ghi trên bị nhiễu quá lớn.

5. KẾT LUẬN

Bài báo đưa ra một phương pháp xây dựng Ha-T2FLS từ dữ liệu và thử nghiệm. Điểm khác biệt của HaT2-FLS với T1-FLS là các tập mờ trong phần giả thiết và phần kết luận của mỗi luật là các HaT2FS, nhờ đó hệ lôgic mờ này có khả năng mô hình hóa tốt với dữ liệu có nhiễu. Hơn nữa, việc có thêm các tham số tính mờ trong ĐSGT, $fm(c^-)$ và $\mu(h_j)$, đã làm cho quá trình tối ưu hệ mờ này có lợi thế hơn T1-FLS. Kết quả thử nghiệm trong bài toán dự báo trên đây là khá tin cậy (LSE = 1.75), điều đó cho thấy tính hợp lý của phương pháp đề xuất cũng như khả năng ứng dụng của HaT2-FLS. Trong tương lai, có thể thử nghiệm hệ mờ này trong những ứng dụng khác.

TÀI LIỆU THAM KHẢO

[1] T. Đ. Khang, Đ. K. Dũng, Suy diễn với tập mờ loại hai dựa trên đại số gia tử, *Tạp chí Tin học và Điều khiển học* **19** (1) (2003) 28–43.

[2] P. A. Phong, T. Đ. Khang, Đ. K. Đông, Mô hình hoá mờ với tập mờ loại hai dựa trên đại số gia tử, *Kỷ yếu Hội thảo Khoa học Quốc gia ICT.rda'08*, Hà Nội, 2008 (103–112).

[3] P. A. Phong, T. Đ. Khang, Biểu diễn tập mờ loại hai đại số gia tử, *Tạp chí Tin học và Điều khiển học* **26** (1) (2010) 29–43.

[4] P. A. Phong, T. Đ. Khang, Đ. K. Đông, Các phép toán trên tập mờ loại hai đại số gia tử, *Kỷ yếu Hội thảo Khoa học Quốc gia FAIR 2009*, Hà Nội, 2010 (33–43).

[5] J. C. Bezdek, A physical interpretation of Fuzzy ISODATA, *IEEE Trans. Syst., Man, Cybern.* **SMC-6** (1976) 387—390.

[6] O. Castillo and P. Melin, *Type-2 fuzzy logic Theory and Applications*, STUDFUZZ 223, Springer-Verlag Berlin Heidelberg, 2008.

[7] O. Cordon, F. Herrera, F. Hoffmann and L. Magdalena, Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases, *Advances in Fuzzy Systems - Applications and Theory* **19**, World Sci., Singapore, (2001).

[8] N. C. Ho, T.D. Khang, N.H. Chau and H.V. Nam, Hedge Algebras, linguistic-valued logic and their application to fuzzy reasoning, *International Journal of Uncertainty, Fuzziness and Knowledge-Based System* **7** (4) (1999) 347–361.

[9] N. C. Ho and W. Wechler, Hedge algebra: An algebraic approach to structures of sets of linguistic truth values, *Fuzzy Sets and Systems* **35** (1990) 281–293.

[10] R. I. John, "Perception Modelling using Type-2 Fuzzy Sets", PhD. thesis, De Montfort University, Leicester, England, 2000.

[11] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: Introduction and new Directions*, Upper Saddle River, NJ Prentice-Hall, 2001.

[12] N. R. Pal and J. C. Bezdek, On cluster validity for fuzzy c-means model, *IEEE Trans. Fuzzy Systems* **1** (1995) 370–379.

[13] P. A. Phong, D. K. Dong and T. D. Khang, Ha-T2FLS and its application to predict survival time of Myeloma Patients, *Proc. of KSE*, Vietnam, 2009 (13–18).

[14] Y. Qiu, Y. Q. Zhang and Y. Zhao, Statistical genetic interval-valued fuzzy systems with prediction in clinical trials, *IEEE Conf. on Granular Computing* (2007) 129–132.

[15] H. Tahayori, A. Tettamanzi, and G. Antoni, Approximated type-2 fuzzy set operations, *Proc. FUZZ-IEEE 2006*, Vancouver, Canada, 2006 (1910–1917).

[16] T. D. Khang, P. A. Phong, D. K. Dong and C. M. Trang, Hedge Algebraic type 2 fuzzy sets, *Proceedings of FUZZ-IEEE 2010*, Barcelona, Spain,July, 2010 (1850–1857).

[17] K. L. Wu, Parameter selections of fuzzy c-means based on robust analysis, *World Academy of Science, Engineering and Technology* **65** (2010) 554–557.

[18] R. R. Yager, Fuzzy subsets of type II in decisions, *Journal of Cybernetics* **10** (1980) 137–159.

[19] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part I, *Information Sciences* **8** (1975) 199–249.

[20] *Database SAS/STAT User Guide 9.2 The PHREG Procedure*, 2008.

Ngày nhận bài 16 - 8 - 2010
Nhận lại sau sửa 30 - 5 - 2011