Thư ngỏ

***Kính gửi quý độc giả Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng công nghệ thông tin và truyền thông*.**

Năm 2020, ngành Thông tin và Truyền thông tập trung thúc đẩy việc chuyển đổi số. Tạp chí Thông tin và Tuyền thông và ấn phẩm khoa học Chuyên san có nhiều đổi mới để các công trình nghiên cứu khoa học trong lĩnh vực CNTT và truyền thông thực sự góp phần đáng kể vào công cuộc chuyển đổi này.

Phát huy truyền thống là một tạp chí khoa học uy tín hàng đầu quốc gia trong lĩnh vực công nghệ thông tin và truyền thông, bước ra hội nhập quốc tế, Chuyên san đặt mục tiêu được vào danh mục trích dẫn tạp chí của các nước ASEAN (ACI) trước năm 2022 và mục tiêu dài hạn vào danh mục SCOPUS.

Để thực hiện mục tiêu đó, tháng 6 vừa qua Tạp chí đã thành lập Thường trực hội đồng biên tập bao gồm 10 nhà khoa học ở các lĩnh vực, đến từ nhiều cơ sở đào tạo và quốc gia khác nhau: Việt Nam, Đài Loan, Anh, Ý, với Chủ tịch là GS.TS. Lê Hoài Bắc.

Trong thời gian tới, Chuyên san sẽ tiếp tục triển khai quy trình bình duyệt bài theo tiêu chuẩn quốc tế một cách nhanh chóng đáp ứng mong muốn về tiến độ của tác giả gửi bài.

Bên cạnh đó, Chuyên san cũng xây dựng chương trình hỗ trợ cụ thể cho các tác giả và nhất là các bạn nghiên cứu sinh thông qua việc tổ chức hội thảo, xây dựng tài liệu hướng dẫn, trợ giúp chuyển đổi ngôn ngữ sang tiếng Anh (nếu cần). Hội đồng biên tập sẽ tích cực tham gia, tài trợ các hội nghị khoa học, phát hiện các nội dung nghiên cứu phát triển tốt có thể gửi bài công bố kết quả trên Chuyên san.

Nhận dịp này, cho phép tôi một lần nữa gửi lời cảm ơn tới các thế hệ Ban Biên tập (nay là Hội đồng biên tập) Chuyên san các thời kỳ trong 21 năm qua, đóng góp vào sự trưởng thành hôm nay của Chuyên san nói riêng và Tạp chí Thông tin và Truyền thông nói chung.

Tổng Biên tập

Nguyễn Văn Bá

Các công trình nghiên cứu phát triển và ứng dụng

Công nghệ Thông tin và Truyền thông

http://www.ictmag.vn

Tập 2020, Số 1

Tháng 6

MỤC LỤC

Thuật toán tìm kiếm Hill climbing giải bài toán Cây Steiner nhỏ nhất

Trần Việt Chương[1], Phan Tấn Quốc[2], Hà Hải Nam[3]

[1] Trung tâm Công nghệ thông tin và Truyền thông, Sở Thông tin và Truyền thông tỉnh Cà Mau

[2] Khoa Công nghệ thông tin, Trường Đại học Sài Gòn

[3] Viện Khoa học Kỹ thuật Bưu điện, Học viện Công nghệ Bưu chính Viễn thông

Tác giả liên hệ: Trần Việt Chương, chuongcm74@gmail.com

Ngày nhận bài: 26/10/2019, ngày sửa chữa: 06/03/2020

Định danh DOI: 10.32913/mic-ict-research-vn.v2020.n1.897

Tóm tắt: Cây Steiner nhỏ nhất (Steiner Minimum Tree-SMT) là bài toán tối ưu tổ hợp có nhiều ứng dụng quan trọng trong khoa học và kỹ thuật. Đây là bài toán thuộc lớp NP-hard. Trước đây đã có nhiều công trình nghiên cứu theo các hướng tiếp cận khác nhau đưa ra các thuật toán để giải bài toán SMT. Bài báo này đề xuất thuật toán Hill climbing search để giải bài toán Cây Steiner nhỏ nhất, trong đó đề xuất cách thức tìm kiếm lân cận tất định và cách thức kết hợp tìm kiếm lân cận tất định với tìm kiếm lân cận ngẫu nhiên để giải quyết bài toán Cây Steiner nhỏ nhất. Kết quả thực nghiệm với hệ thống dữ liệu thực nghiệm chuẩn cho thấy thuật toán đề xuất của chúng tôi cho lời giải với chất lượng tốt hơn so với một số thuật toán heuristic hiện biết trên một số bộ dữ liệu.

Từ khóa: *thuật toán tìm kiếm leo đồi, Cây Steiner nhỏ nhất, thuật toán heuristic, thuật toán metaheuristic, lân cận tất định, lân cận ngẫu nhiên, tối ưu cục bộ.*

Title: **Hill Climbing Search Algorithm For Solving Steiner Minimum Tree Problem**

Abstract: Steiner Minimum Tree (SMT) is a complex optimization problem that has many important applications in science and technology. This is a NP-hard problem. In the past, there were many research projects based on different approaches offering algorithms to solve SMT problems. This paper proposes a Hill climbing search algorithm to solve the SMT problem; which suggests how to search nearby and how to combine with random nearby searches to solve local optimization problems. Experimental results with the standard experimental data system show that our proposed algorithm offers better quality solution than some existing heuristic algorithms on some data sets.

Keywords: *Hill climbing search algorithm, Steiner minimum tree problem, heuristic algorithm, metaheuristic algorithm, neighboring deterministic, random nearby, local optimization.*

I. GIỚI THIỆU

Giả sử chúng ta cần xây dựng một hệ thống giao thông nối một số địa điểm trong khu đô thị. Vấn đề đặt ra là phải thiết kế sao cho hệ thống giao thông đó có độ dài ngắn nhất nhằm giảm kinh phí xây dựng. Yêu cầu của bài toán cho phép thêm vào một số địa điểm trung gian để giảm thiểu tổng chiều dài hệ thống cần xây dựng. Đây là bài toán có thể vận dụng kiến thức bài toán Cây Steiner nhỏ nhất để giải.

Bài toán Cây Steiner nhỏ nhất hiện được nghiên cứu ở một số dạng sau: Bài toán thứ nhất là bài toán Cây Steiner nhỏ nhất với khoảng cách Euclide [11, 27, 28]. Bài toán thứ hai là bài toán Cây Steiner nhỏ nhất với khoảng cách chữ nhật [11]. Bài toán thứ ba là Cây Steiner nhỏ nhất với khoảng cách được cho ngẫu nhiên [4]. Đây là giới hạn nghiên cứu về bài toán Cây Steiner nhỏ nhất của chúng tôi trong bài báo này [28].

Mục này trình bày một số định nghĩa và khả năng ứng dụng của bài toán Cây Steiner nhỏ nhất.

A. Một số định nghĩa

Định nghĩa 1: Cây Steiner [4]

Cho $G = (V(G), E(G))$ là một đơn đồ thị vô hướng liên thông và có trọng số không âm trên cạnh, $V(G)$ là tập gồm n đỉnh, $E(G)$ là tập gồm m cạnh, $w(e)$ là trọng số của cạnh e với $e \in E(G)$. Cho $L \subseteq V(G)$, cây T đi qua tất cả các đỉnh trong tập L, tức với $L \subseteq V(T)$ được gọi là Cây Steiner của L.

Tập L được gọi là tập *terminal*, các đỉnh thuộc tập L được gọi là đỉnh *terminal*.

Đỉnh thuộc cây T mà không thuộc tập L được gọi là đỉnh *Steiner* của cây T đối với tập L.

Định nghĩa 2: Chi phí Cây Steiner [4]

Cho $T = (V(T), E(T))$ là một Cây Steiner của đồ thị G. Chi phí của cây T, ký hiệu là $C(T)$, là tổng trọng số của các cạnh thuộc cây T, tức là ta có $C(T) = \sum_{e \in E(T)} w(e)$.

Định nghĩa 3: Cây Steiner nhỏ nhất [4]

Cho đồ thị G được mô tả như trên, bài toán tìm Cây Steiner có chi phí nhỏ nhất được gọi là bài toán Cây Steiner nhỏ nhất (Steiner Minimum Tree problem - SMT) hoặc được gọi ngắn gọn là bài toán Cây Steiner (Steiner Tree problem).

SMT là bài toán tối ưu tổ hợp của lý thuyết đồ thị. Trong trường hợp tổng quát, SMT đã được chứng minh thuộc lớp *NP-hard* [4, 13].

Khác với bài toán cây khung nhỏ nhất (Minimum Spanning Trees Problem) - đó là bài toán đơn giản. Cây Steiner là cây đi qua tất cả các đỉnh thuộc tập *terminal* L và có thể thêm một số đỉnh khác nữa thuộc tập $V(G)$ chứ không nhất thiết phải đi qua tất cả các đỉnh của đồ thị.

Để ngắn gọn, trong bài báo này từ *đồ thị* sẽ được hiểu là đơn đồ thị, vô hướng, liên thông và có trọng số không âm trên cạnh.

Định nghĩa 4: *1-lân cận* của Cây Steiner T

Cho đồ thị G và T là một Cây Steiner của G. Ta gọi *1-lân cận* của Cây Steiner T là tập tất cả các Cây Steiner của đồ thị G sai khác với T đúng một cạnh. Nếu T' là một Cây Steiner thuộc *1-lân cận* của T thì ta nói T và T' là *1-lân cận* với nhau.

Trong một số trường hợp chúng ta còn sử dụng những lân cận rộng hơn so với *1-lân cận*. Khái niệm *k-lân cận* dưới đây là mở rộng trực tiếp của khái niệm *1-lân cận*.

Định nghĩa 5: *k-lân cận* của Cây Steiner T

Cho đồ thị G và T là một Cây Steiner của nó. Ta gọi *k-lân cận* của Cây Steiner T là tập tất cả các Cây Steiner của đồ thị G sai khác với T không quá k cạnh. Nếu T' là một Cây Steiner thuộc *k-lân cận* của T thì ta nói T và T' là *k-lân cận* với nhau.

Định nghĩa 6: Lân cận tất định và lân cận ngẫu nhiên

Nếu các Cây Steiner trong lân cận được xác định không phụ thuộc vào yếu tố ngẫu nhiên thì ta nói về lân cận tất định, còn nếu ngược lại, ta nói về lân cận ngẫu nhiên.

B. Ứng dụng của bài toán Cây Steiner nhỏ nhất

Bài toán SMT có thể được tìm thấy trong các ứng dụng quan trọng như: Bài toán thiết kế mạng truyền thông, bài toán thiết kế VLSI (Very Large Scale Integrated), các bài toán liên quan đến hệ thống mạng với chi phí nhỏ nhất [3, 4, 12, 17, 31].

Đóng góp chính của chúng tôi trong bài báo này là đã đề xuất cách thức tìm kiếm lân cận mới để giải bài toán Cây Steiner nhỏ nhất đồng thời cài đặt thực nghiệm thuật toán trên hệ thống dữ liệu thực nghiệm chuẩn.

II. KHẢO SÁT MỘT SỐ THUẬT TOÁN GIẢI BÀI TOÁN CÂY STEINER NHỎ NHẤT

Hiện tại, có nhiều hướng tiếp cận giải bài toán Cây Steiner nhỏ nhất như các thuật toán rút gọn đồ thị, các thuật toán tìm lời giải đúng, các thuật toán tìm lời giải gần đúng cận tỉ lệ, các thuật toán heuristic và các thuật toán metaheuristic.

A. Các thuật toán rút gọn đồ thị

Một số nghiên cứu trình bày các kỹ thuật nhằm giảm thiểu kích thước của đồ thị. Chẳng hạn công trình của Jeffrey H.Kingston và Nicholas Paul Sheppard [10], công trình của Thorsten Koch Alexander Martin [26], C. C. Ribeiro, M.C. Souza [5],...

Ý tưởng chung của các thuật toán rút gọn đồ thị là nhằm gia tăng các đỉnh cho tập terminal và loại bỏ các đỉnh của đồ thị mà nó chắc chắn không thuộc về Cây Steiner nhỏ nhất cần tìm. Chất lượng các thuật toán giải bài toán SMT phụ thuộc vào độ lớn của hệ số $n - |L|$. Do vậy, mục đích của các thuật toán rút gọn đồ thị là làm giảm thiểu tối đa hệ số $n - |L|$.

Các thuật toán rút gọn đồ thị cũng được xem là bước tiền xử lý dữ liệu quan trọng trong việc giải bài toán SMT. Hướng tiếp cận này là rất cần thiết khi tiếp cận bài toán SMT bằng các thuật toán tìm lời giải đúng [16].

B. Các thuật toán tìm lời giải đúng

Các nghiên cứu tìm lời giải đúng cho bài toán SMT như thuật toán quy hoạch động của Dreyfus và Wagner [33], thuật toán dựa trên phép nới lỏng Lagrange của Beasley [9], các thuật toán nhánh cận của Koch và Martin [26], Xinhui Wang [15, 30],...

Ưu điểm của hướng tiếp cận này là tìm được lời giải chính xác, nhược điểm của hướng tiếp cận này là chỉ giải được các bài toán có kích thước nhỏ; nên khả năng ứng dụng của chúng không cao. Việc giải đúng bài toán SMT thực sự là một thách thức trong lý thuyết tối ưu tổ hợp.

Hướng tiếp cận này là cơ sở quan trọng có thể được sử dụng để đánh giá chất lượng lời giải của các thuật toán giải gần đúng khác khi giải bài toán SMT.

C. Các thuật toán gần đúng cận tỉ lệ

Thuật toán gần đúng cận tỉ lệ α nghĩa là lời giải tìm được gần đúng một cận tỉ lệ α so với lời giải tối ưu.

Ưu điểm của thuật toán gần đúng cận tỉ lệ là có sự đảm bảo về mặt toán học theo nghĩa cận tỉ lệ trên, nhược điểm của thuật toán dạng này là cận tỉ lệ tìm được trong thực tế thường là kém hơn nhiều so với chất lượng lời giải tìm được bởi nhiều thuật toán gần đúng khác dựa trên thực nghiệm.

Thuật toán MST-Steiner của Bang Ye Wu và Kun-Mao Chao có cận tỉ lệ 2 [4], thuật toán Zelikovsky-Steiner có cận tỉ lệ 11/6 [4]. Cận tỉ lệ tốt nhất tìm được hiện tại cho bài toán SMT là 1.39 [6, 18, 24].

D. Các thuật toán heuristic

Thuật toán heuristic chỉ ra những kinh nghiệm riêng biệt để tìm kiếm lời giải cho một bài toán tối ưu cụ thể. Thuật toán heuristic thường tìm được lời giải có thể chấp nhận được trong thời gian cho phép nhưng không chắc đó là lời giải chính xác. Các thuật toán heuristic cũng không chắc hiệu quả trên mọi loại dữ liệu đối với một bài toán cụ thể.

Các thuật toán heuristic điển hình cho bài toán SMT như: SPT [21], PD Steiner [21] (bài báo này sẽ gọi tắt là PD), SPH [5], Heu [26], Distance network heuristic của Kou, Markowsky và Berman [34].

Ưu điểm của các thuật toán heuristic là thời gian chạy nhanh hơn nhiều so với các thuật toán metaheuristic. Giải pháp này thường được lựa chọn với các đồ thị có kích thước lớn [11, 20, 21, 25].

E. Các thuật toán metaheuristic

Thuật toán metaheuristic sử dụng nhiều heuristic kết hợp với các kỹ thuật phụ trợ nhằm khai phá không gian tìm kiếm, metaheuristic thuộc lớp các thuật toán tìm kiếm tối ưu.

Hiện đã có nhiều công trình sử dụng thuật toán metaheuristic giải bài toán SMT. Chẳng hạn như thuật toán local search sử dụng các chiến lược tìm kiếm lân cận node-based neighborhood [19], path-based neighborhood [19], thuật toán local search [7], thuật toán tìm kiếm lân cận biến đổi [23], thuật toán di truyền [2], thuật toán tabu search [5], thuật toán di truyền song song [14], thuật toán bees [22],...

Từ những hướng tiếp cận trên, bài báo này đề xuất một thuật toán metaheuristic dạng cá thể. Cụ thể là thuật toán Hill climbing search để giải bài toán SMT. Cách tiếp cận này có thể giải được các bài toán SMT có kích thước lớn, có chất lượng tốt hơn so với các hướng tiếp cận heuristic, các thuật toán gần đúng cận tỉ lệ và có thời gian chạy nhanh hơn so với các thuật toán metaheuristic dạng quần thể [32].

III. THUẬT TOÁN HILL CLIMBING SEARCH

A. Ý tưởng thuật toán Hill climbing search

Thuật toán Hill climbing search là một trong những kỹ thuật dùng để tìm kiếm tối ưu cục bộ cho một bài toán tối ưu [1].

Thuật toán Hill climbing search là một trong những giải pháp để giải bài toán tối ưu, đặc biệt là dạng bài toán ưu tiên về thời gian tính như bài toán dạng thiết kế.

B. Sơ đồ tổng quát thuật toán Hill climbing search

Thuật toán Hill climbing search liên tục thực hiện việc di chuyển từ một lời giải S đến một lời giải mới S' trong một cấu trúc lân cận xác định trước theo sơ đồ sau:

Bước 1: Khởi tạo. Chọn lời giải xuất phát S, tính giá trị hàm mục tiêu $F(S)$.

Bước 2: Sinh lân cận. Chọn tập lân cận $N(S)$ và tìm lời giải S' trong tập lân cận này với giá trị hàm mục tiêu $F(S')$.

Bước 3: Test chấp nhận. Kiểm tra xem có chấp nhận di chuyển từ S sang S'. Nếu chấp nhận thì thay S bằng S'; trái lại giữ nguyên S là lời giải hiện tại.

Bước 4: Test điều kiện dừng. Nếu điều kiện dừng thỏa mãn thì kết thúc thuật toán và đưa ra lời giải tốt nhất tìm được; trái lại quay lại bước 2.

C. Giải quyết vấn đề tối ưu hóa cục bộ

Vấn đề lớn nhất mà thuật toán Hill climbing search gặp phải là nó dễ rơi vào bẫy tối ưu cục bộ, đó là lúc chúng ta leo lên một đỉnh mà chúng ta không thể tìm được một lân cận nào tốt hơn được nữa nhưng đỉnh này lại không phải là đỉnh cao nhất.

Để giải quyết vấn đề này, khi leo đến một đỉnh tối ưu cục bộ, để tìm được lời giải tốt hơn nữa ta có thể lặp lại thuật toán Hill climbing search với nhiều điểm xuất phát khác nhau được chọn ngẫu nhiên và lưu lại kết quả tốt nhất ở mỗi lần lặp. Nếu số lần lặp đủ lớn thì ta có thể tìm đến được đỉnh tối ưu toàn cục, tuy nhiên với những bài toán mà không gian tìm kiếm lớn; thì việc tìm được lời giải tối ưu toàn cục là một thách thức lớn [1, 29].

Để giải quyết vấn đề tối ưu cục bộ, trong bài báo này chúng tôi đề xuất việc kết hợp thuật toán Hill climbing search với chiến lược tìm kiếm lân cận ngẫu nhiên để hy vọng nâng cao chất lượng lời giải của thuật toán.

IV. THUẬT TOÁN HILL CLIMBING SEARCH GIẢI BÀI TOÁN CÂY STEINER NHỎ NHẤT

A. Ý tưởng thuật toán

Cho đồ thị vô hướng liên thông có trọng số G. Bắt đầu từ Cây Steiner T của G được khởi tạo ngẫu nhiên, chèn lần lượt từng cạnh $e \in E(G) - E(T)$ vào Cây Steiner T. Nếu Cây Steiner T không chứa chu trình thì cạnh e không cần được xem xét; nếu $E(T) \cup e$ chứa một chu trình thì tìm một cạnh e' trên chu trình này sao cho việc loại nó dẫn đến Cây Steiner T' có chi phí nhỏ nhất. Tiếp theo, nếu $C(T') < C(T)$ thì thay T bằng T'. Thuật toán dừng nếu trong một lần duyệt qua tất cả các cạnh $e \in E(G) - E(T)$ mà không cải thiện được chi phí của Cây Steiner T.

Chúng tôi đặt tên thuật toán Hill climbing search giải bài toán SMT là HCSMT.

B. Sơ đồ thuật toán HCSMT

Thuật toán 1: Thuật toán HCSMT

Đầu vào: Đồ thị $G = (V(G), E(G))$
Đầu ra : Cây Steiner có chi phí nhỏ nhất tìm được

```
1  stop = false;
2  d = 0; //d là số lần tăng cường
3  T là Cây Steiner được khởi tạo ngẫu nhiên;
4  T_best = T; //lưu lại Cây Steiner tốt nhất trước khi thực
     hiện chiến lược đa dạng hóa;
5  while (d < N){ // N là số lần lặp định trước
6      stop = true;
7      S = E − E(T);
8      for (với mỗi cạnh e ∈ S){
9          min = +∞;
10         if (cạnh e có 2 đỉnh (u, v) ∈ T){
11             F = T ∪ e;
12             Xác định chu trình Cycle trong F chứa e;
13             for (với mỗi cạnh e' ∈ Cycle)
14                 if (C(F − e') < min){
15                     min = C(F − e');
16                     Ghi nhận T' = F − e';
17                 }
18             if (min < C(T)){
19                 T = T';
20                 stop = false;
21             }
22         }
23     }
24     if (stop){
25         d++; //tăng số biến đếm tăng cường
26         if (C(T) < C(T_best)) T_best = T;
27         while (Thỏa điều kiện đa dạng hóa){
28             Loại bỏ một số cạnh ngẫu nhiên của Cây
                 Steiner đang xét;
29             Chọn ngẫu nhiên một số cạnh trong các
                 cạnh còn lại của đồ thị G cho đến khi cây
                 T liên thông trở lại (điều kiện cạnh thêm
                 vào là: Phải có đỉnh thuộc T mới được
                 thêm vào để tránh trường hợp thêm vào
                 quá nhiều lần nhưng T không liên thông trở
                 lại). Sử dụng định nghĩa k-lân cận ở trên;
30             Nếu thêm quá nhiều lần mà không làm cho
                 T liên thông trở lại, thì tiếp tục đa dạng
                 hóa lại với các cạnh khác;
31         }
32     }
33 }
34 return T_best;
```

C. Một số bàn luận thêm

a) Vấn đề tạo lời giải ngẫu nhiên ban đầu

Cũng lưu ý rằng Cây Steiner ban đầu được khởi tạo ngẫu nhiên thông qua hai giai đoạn như sau (dòng 3 của thuật toán HCSMT):

Bắt đầu từ cây chỉ gồm một đỉnh nào đó của đồ thị, tiếp theo, thuật toán sẽ thực hiện $n-1$ bước lặp với n là số đỉnh của đồ thị đang xét. Ở mỗi bước lặp, trong số các đỉnh chưa được chọn để tham gia vào cây, ta chọn một đỉnh kề với ít nhất một đỉnh nằm trong cây đang được xây dựng mà không quan tâm đến trọng số của cạnh. Đỉnh được chọn và cạnh nối nó với đỉnh của cây đang được xây dựng sẽ được bổ sung vào cây; thuật toán này sử dụng ý tưởng của thuật toán Prim.

Duyệt các đỉnh treo $u \in T$, nếu $u \notin L$ thì xóa cạnh chứa đỉnh u khỏi $E(T)$, xóa đỉnh u trong $V(T)$ và cập nhật bậc của đỉnh kề với đỉnh u trong T. Lặp lại bước này đến khi T không còn sự thay đổi nào nữa; bước này gọi là bước xóa các cạnh dư thừa.

b) Vấn đề tìm kiếm lân cận

Thao tác tìm chu trình trong Cây Steiner T sau khi chèn thêm một cạnh e được tiến hành như sau: Khi chèn cạnh $e = (u, v)$ vào T, duyệt Cây Steiner T theo chiều sâu bắt đầu từ u, lưu vết trên đường đi bằng mảng p (đỉnh trước của một đỉnh trong phép duyệt). Tiếp theo, bắt đầu từ đỉnh v, truy vết theo mảng p đến khi gặp u thì kết thúc, các cạnh trên đường truy vết chính là các cạnh trong chu trình cần tìm.

Thuật toán HCSMT ngoài việc lời giải ban đầu được khởi tạo ngẫu nhiên thì các Cây Steiner lân cận tìm được trong quá trình tìm kiếm là kiểu *1-lân cận* tất định. Hiệu quả của thuật toán HCSMT có thể được cải thiện khi ta thay đổi thứ tự các cạnh được duyệt trong tập $E(G) - E(T)$; nghĩa là ta sẽ duyệt tập cạnh này theo một hoán vị được sinh ngẫu nhiên chứ không theo một thứ tự cố định ở tất cả các lần duyệt.

c) Vấn đề kết hợp với tìm kiếm ngẫu nhiên

Thuật toán HCSMT chủ yếu sử dụng tính tăng cường, thể hiện qua các chiến lược tìm kiếm Cây Steiner lân cận. Tính đa dạng được sử dụng vào các thời điểm sau: thứ nhất là khi khởi tạo lời giải ban đầu, thứ hai là thay đổi thứ tự duyệt của các cạnh trong tập cạnh ứng viên (như đã đề cập ở đoạn trên), thứ ba khi việc tìm kiếm lân cận không cải thiện qua một số lần lặp thì tiến hành chọn ngẫu nhiên một số cạnh của cây để bắt đầu trở lại việc tìm kiếm lân cận.

V. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Phần này chúng tôi mô tả chi tiết việc thực nghiệm thuật toán HCSMT do chúng tôi đề xuất; đồng thời đưa ra một số so sánh, đánh giá về các kết quả đạt được.

Bảng I
KẾT QUẢ THỰC NGHIỆM THUẬT TOÁN TRÊN NHÓM ĐỒ THỊ STEINC

Test	*n*	*m*	Heu	PD	VNS	TS	HC SMT
steinc1.txt	500	625	85	85	85	85	85
steinc2.txt	500	625	144	144	144	144	144
steinc3.txt	500	625	755	762	754	754	754
steinc4.txt	500	625	1080	1085	1079	1079	1080
steinc5.txt	500	625	1579	1583	1579	1579	1579
steinc6.txt	500	1000	55	55	55	55	55
steinc7.txt	500	1000	102	102	102	102	102
steinc8.txt	500	1000	510	516	509	509	509
steinc9.txt	500	1000	715	718	707	707	707
steinc10.txt	500	1000	1093	1107	1093	1093	1093
steinc11.txt	500	2500	32	34	33	32	32
steinc12.txt	500	2500	46	48	46	46	46
steinc13.txt	500	2500	262	268	258	258	258
steinc14.txt	500	2500	324	332	323	324	324
steinc15.txt	500	2500	557	562	556	556	557
steinc16.txt	500	12500	11	12	11	11	11
steinc17.txt	500	12500	19	20	18	18	18
steinc18.txt	500	12500	120	123	115	117	115
steinc19.txt	500	12500	150	159	148	148	149
steinc20.txt	500	12500	268	268	268	267	268

A. Dữ liệu thực nghiệm

Để thực nghiệm các thuật toán liên quan, chúng tôi sử dụng 60 bộ dữ liệu là các đồ thị thưa trong hệ thống dữ liệu thực nghiệm chuẩn dùng cho bài toán Cây Steiner tại địa chỉ URL http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html[8]. Trong đó nhóm steinc có 20 đồ thị, nhóm steind có 20 đồ thị, nhóm steine có 20 đồ thị (các đồ thị này có tập $|L|$ tối đa 1250 đỉnh).

B. Môi trường thực nghiệm

Thuật toán HCSMT được chúng tôi cài đặt bằng ngôn ngữ C++ sử dụng môi trường DEV C++ 5.9.2; được thực nghiệm trên một máy chủ ảo, Hệ điều hành Windows server 2008 R2 Enterprise 64bit, Intel(R) Xeon (R) CPU E5-2660 @ 2,20 GHz, RAM 4GB.

Bảng II
KẾT QUẢ THỰC NGHIỆM THUẬT TOÁN TRÊN NHÓM ĐỒ THỊ STEIND

Test	*n*	*m*	Heu	PD	VNS	TS	HC SMT
steind1.txt	1000	1250	106	107	106	106	106
steind2.txt	1000	1250	220	228	220	220	220
steind3.txt	1000	1250	1570	1771	1565	1567	1565
steind4.txt	1000	1250	1936	2174	1935	1935	1936
steind5.txt	1000	1250	3252	3511	3250	3250	3250
steind6.txt	1000	2000	70	70	67	70	67
steind7.txt	1000	2000	103	111	103	103	103
steind8.txt	1000	2000	1092	1287	1073	1078	1073
steind9.txt	1000	2000	1462	1773	1448	1450	1448
steind10.txt	1000	2000	2113	2550	2111	2112	2113
steind11.txt	1000	5000	29	29	29	30	29
steind12.txt	1000	5000	42	44	42	42	42
steind13.txt	1000	5000	510	643	502	502	507
steind14.txt	1000	5000	675	851	671	667	674
steind15.txt	1000	5000	1120	1437	1116	1117	1118
steind16.txt	1000	25000	13	13	13	13	13
steind17.txt	1000	25000	23	25	23	23	23
steind18.txt	1000	25000	238	301	228	230	231
steind19.txt	1000	25000	325	424	318	315	321
steind20.txt	1000	25000	539	691	538	538	539

C. Kết quả thực nghiệm và đánh giá

Kết quả thực nghiệm của các thuật toán được ghi nhận ở các Bảng I,II,III. Các bảng này có cấu trúc như sau: Cột đầu tiên (Test) là tên các bộ dữ liệu trong hệ thống dữ liệu thực nghiệm, số đỉnh (n), số cạnh (m) của từng đồ thị, các cột tiếp theo ghi nhận giá trị chi phí Cây Steiner lần lượt ứng với hai thuật toán hueristic: Heu [26], PD [21]. Hai thuật toán metahueristic: VNS [23], TS [5] và thuật toán HCSMT.

Bộ tham số được xác định như sau qua thực nghiệm: Số lần chạy mỗi bộ dữ liệu là 30, số lần tăng cường là 50; số cạnh loại bỏ ngẫu nhiên của mỗi lần tăng cường là $0{,}05\times|E(T)|$.

D. Đánh giá kết quả thực nghiệm

Mục này nhằm so sánh chất lượng lời giải của thuật toán HCSMT với hai thuật toán heuristic: Heu [26], PD [21] và hai thuật toán metaheuristic: VNS [23], TS [5]. Nội dung của Bảng IV,V,VI,VII cho biết số lượng (SL) và tỉ lệ phần trăm (%) tương ứng với số lượng bộ dữ liệu cho chất lượng lời giải tốt hơn (ghi nhận bởi ký hiệu "<") hoặc cho chất lượng lời giải bằng nhau (ghi nhận bởi ký hiệu "=") hoặc

Bảng III
KẾT QUẢ THỰC NGHIỆM THUẬT TOÁN TRÊN NHÓM ĐỒ THỊ STEINE

Test	*n*	*m*	Heu	PD	VNS	TS	HCSMT
steine1.txt	2500	3125	111	111	111	111	111
steine2.txt	2500	3125	214	214	214	216	214
steine3.txt	2500	3125	4052	4570	4015	4018	4015
steine4.txt	2500	3125	5114	5675	5101	5105	5101
steine5.txt	2500	3125	8130	8976	8128	8128	8130
steine6.txt	2500	5000	73	73	73	73	73
steine7.txt	2500	5000	149	150	145	149	145
steine8.txt	2500	5000	2686	3254	2648	2649	2648
steine9.txt	2500	5000	3656	4474	3608	3605	3608
steine10.txt	2500	5000	5614	6847	5600	5602	5600
steine11.txt	2500	12500	34	34	34	34	34
steine12.txt	2500	12500	68	68	67	68	68
steine13.txt	2500	12500	1312	1704	1292	1299	1312
steine14.txt	2500	12500	1752	2304	1735	1740	1735
steine15.txt	2500	12500	2792	3626	2784	2784	2799
steine16.txt	2500	62500	15	15	15	15	15
steine17.txt	2500	62500	26	27	25	25	25
steine18.txt	2500	62500	608	804	583	595	594
steine19.txt	2500	62500	788	1059	768	778	768
steine20.txt	2500	62500	1349	1753	1342	1352	1342

cho chất lượng lời giải kém hơn (ghi nhận bởi ký hiệu ">") khi so sánh thuật toán HCSMT với các thuật toán: Heu [26], PD [21], VNS [23] và TS [5].

Với 20 bộ dữ liệu nhóm steinc, thuật toán HCSMT cho chất lượng lời giải (tốt hơn, bằng, kém hơn) thuật toán Heu [26] (35,0%, 65,0%, 0,0%). Kết quả so sánh thuật toán HCSMT với thuật toán Heu [26] trên các nhóm dữ liệu steind, steine cũng đã được thể hiện chi tiết ở Bảng IV.

Bảng IV
SO SÁNH CHẤT LƯỢNG LỜI GIẢI CỦA THUẬT TOÁN HCSMT VỚI THUẬT TOÁN HEU

Nhóm đồ thị	HCSMT<Heu		HCSMT=Heu		HCSMT>Heu	
	SL	%	SL	%	SL	%
steinc	7	35%	13	65%	0	0%
steind	10	50%	10	50%	0	0%
steine	11	55%	8	40%	1	5%
Tổng cộng:	**28**	**46,7%**	**31**	**51,7%**	**1**	**1,6%**

Tương tự, kết quả so sánh thuật toán HCSMT với thuật toán PD [21] trên các nhóm dữ liệu steinc, steind, steine cũng đã được thể hiện chi tiết ở Bảng V.

Bảng V
SO SÁNH CHẤT LƯỢNG LỜI GIẢI CỦA THUẬT TOÁN HCSMT VỚI THUẬT TOÁN PD

Nhóm đồ thị	HCSMT<PD		HCSMT=PD		HCSMT>PD	
	SL	%	SL	%	SL	%
steinc	15	75%	5	25%	0	0%
steind	18	90%	2	10%	0	0%
steine	14	70%	6	30%	0	0%
Tổng cộng:	**47**	**78,4%**	**13**	**21,6%**	**0**	**0%**

Kết quả so sánh thuật toán HCSMT với thuật toán VNS [23] trên các nhóm dữ liệu steinc, steind, steine cũng đã được thể hiện chi tiết ở Bảng VI.

Đánh giá chung trên toàn bộ 60 bộ dữ liệu, thuật toán HCSMT cho chất lượng lời giải (tốt hơn, bằng, kém hơn) thuật toán Heu [26] (46,7%, 51,7%, 1,6%). Thuật toán HCSMT cho chất lượng lời giải (tốt hơn, bằng, kém hơn) thuật toán PD [21] (78,4%, 21,6%, 0,0%). Thuật toán HCSMT cho chất lượng lời giải (tốt hơn, bằng, kém hơn) thuật toán VNS [23] (1,7%, 70%, 28,3%). Thuật toán HCSMT cho chất lượng lời giải (tốt hơn, bằng, kém hơn) thuật toán TS [5] (26,7%, 46,6%, 26,7%).

Thời gian tính trung bình của thuật toán HCSMT trên các đồ thị ứng với nhóm dữ liệu steinc là 2,54 giây. Tương tự với nhóm đồ thị steind là 16,78 giây, với nhóm đồ thị steine là 214,81 giây. Thời gian chạy chương trình ngoài việc phụ thuộc vào độ phức tạp thời gian tính của thuật toán, còn phụ thuộc vào môi trường thực nghiệm, kỹ thuật lập trình, cấu trúc dữ liệu chọn cài đặt, các bộ tham số,... và thông thường, thời gian chạy của thuật toán heuristic nhanh hơn thuật toán metaheuristic. Tuy vậy, thời gian chạy của thuật toán HCSMT như trên cũng cho chúng ta thông tin tham khảo cần thiết về thuật toán này.

Bảng VI
SO SÁNH CHẤT LƯỢNG LỜI GIẢI CỦA THUẬT TOÁN HCSMT VỚI THUẬT TOÁN VNS

Nhóm đồ thị	HCSMT<VNS		*HCSMT=VNS*		*HCSMT>VNS*	
	SL	%	SL	%	SL	%
steinc	1	5%	15	75%	4	20%
steind	0	0%	12	60%	8	40%
steine	0	0%	15	75%	5	25%
Tổng cộng:	**1**	**1,7%**	**42**	**70%**	**17**	**28,3%**

Kết quả so sánh thuật toán HCSMT với thuật toán TS [5] trên các nhóm dữ liệu steinc, steind, steine cũng đã được thể hiện chi tiết ở Bảng VII.

Bảng VII
SO SÁNH CHẤT LƯỢNG LỜI GIẢI CỦA THUẬT TOÁN HCSMT VỚI THUẬT TOÁN TS

Nhóm đồ thị	HCSMT<TS		*HCSMT=TS*		*HCSMT>TS*	
	SL	**%**	**SL**	**%**	**SL**	**%**
steinc	1	5%	15	75%	4	20%
steind	5	25%	7	35%	8	40%
steine	10	50%	6	30%	4	20%
Tổng cộng:	**16**	**26,7%**	**28**	**46,6%**	**16**	**26,7%**

VI. KẾT LUẬN

Bài báo này đề xuất thuật toán HCSMT dạng thuật toán Hill climbing search để giải bài toán Cây Steiner nhỏ nhất. Đóng góp của chúng tôi trong bài báo này là đã đề xuất cách thức tìm kiếm lân cận kết hợp với tìm kiếm lân cận ngẫu nhiên nâng cao chất lượng của thuật toán. Chúng tôi đã thực nghiệm thuật toán đề xuất trên hệ thống gồm 60 bộ dữ liệu thực nghiệm chuẩn. Kết quả thực nghiệm cho thấy rằng thuật toán này cho chất lượng lời giải tốt hơn hoặc bằng hai thuật toán dạng heuristic và cho chất lượng lời giải tốt hơn, bằng, kém hơn hai thuật toán dạng metaheuristic tốt hiện biết trên một số bộ dữ liệu thực nghiệm chuẩn.

TÀI LIỆU THAM KHẢO

[1] Alan W. Johnson, *"Generalized Hill Climbing Algorithms For Discrete Optimization Problems"*, doctor of philosophy, Industrial And Systems Engineering, Blacksburg, Virginia, pp.1-119, 1996.

[2] Ankit Anand, Shruti, Kunwar Ambarish Singh, *"An efficient approach for Steiner tree problem by genetic algorithm"*, International Journal of Computer Science and Engineering (SSRG-IJCSE), vol.2, pp.233-237, 2015.

[3] Arie M.C.A. Koster, Xavier Munoz (Eds), *"Graphs and algorithms in communication networks"*, Springer, pp.1-177, 2010.

[4] Bang Ye Wu, Kun-Mao Chao, *"Spanning trees and optimization problems"*, Chapman&Hall/CRC, pp.13-139, 2004.

[5] C. C. RIBEIRO, M.C. SOUZA, *"Tabu Search for the Steiner problem in graphs"*, Networks, 36, pp.138-146, 2000.

[6] Chi-Yeh Chen, *"An efficient approximation algorithm for the Steiner tree problem"*, National Cheng Kung University, Taiwan, 2018.

[7] Eduardo Uchoa, Renato F. Werneck, *"Fast local search for Steiner trees in graphs"*, SIAM, 2010.

[8] J. E. Beasley, OR-Library: URL http://people.brunel.ac.uk/ mastjjb/jeb/orlib/steininfo.html

[9] J. E. Beasley, *"An SST-Based algorithm for the Steiner problem in graphs"*, Networks, 19, pp.1-16, 1989.

[10] Jeffrey H. Kingston, Nicholas Paul Sheppard, *"On reductions for the Steiner problem in graphs"*, Basser Department of Computer Science the University of Sydney, Australia, pp.1-10, 2006.

[11] Jon William Van Laarhoven, *"Exact and heuristic algorithms for the euclidean Steiner tree problem"*, University of Iowa, 2010 (Doctoral thesis).

[12] M. Hauptmann, M. Karpinski (Eds), *"A compendium on Steiner tree problems"*, pp.1-36, 2015.

[13] Marcello Caleffi, Ian F. Akyildiz, Luigi Paura, *"On the solution of the Steiner tree np-hard problem via physarum bionetwork"*, IEEE, pp.1092-1106, 2015.

[14] Nguyen Viet Huy, Nguyen Duc Nghia, *"Solving graphical Steiner tree problem using parallel genetic algorithm"*, RIVF, 2008.

[15] Ondra Suchy, *"Exact algorithms for Steiner tree"*, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic, 2014.

[16] Phan Tấn Quốc, *"Rút gọn đồ thị cho bài toán Cây Steiner nhỏ nhất"*, Kỷ yếu Hội nghị Quốc gia lần thứ IX về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Trường Đại học Cần Thơ, ngày 04-05/08/2016, ISBN: 978-604-913-472-2, trang 638-644, 2016.

[17] Poompat Saengudomlert, *"Optimization for Communications and Networks"*, Taylor and Francis Group, LLC, pp.1-201, 2011.

[18] Reyan Ahmed and et al, *"Multi-level Steiner tree"*, ACM J. Exp. Algor., 1(1), 2018.

[19] S.L. Martins, M.G.C. Resende, C.C. Ribeiro, and P.M. Pardalos, *"A parallel grasp for the Steiner tree problem in graphs using a hybrid local search strategy"*, 1999.

[20] Stefan Hougardy, Jannik Silvanus, Jens Vygen, *"Dijkstra meets Steiner: a fast-exact goal-oriented Steiner tree algorithm"*, Research Institute for Discrete Mathematics, University of Bonn, pp.1-59, 2015.

[21] Trần Việt Chương, Phan Tấn Quốc, Hà Hải Nam, *"Đề xuất một số thuật toán heuristic giải bài toán Cây Steiner nhỏ nhất"*, Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Trường Đại học Đà Nẵng, ngày 17-18/08/2017, ISBN: 978-604-913-614-6, trang 138-147, 2017.

[22] Trần Việt Chương, Phan Tấn Quốc, Hà Hải Nam, *"Thuật toán Bees giải bài toán Cây Steiner nhỏ nhất trong trường hợp đồ thị thưa"*. Tạp chí khoa học công nghệ thông tin và truyền thông, Học viên Công nghệ Bưu chính Viễn thông, Bộ Thông tin và Truyền thông, ISSN: 2525-2224, tháng 12, trang 24-29, 2017.

[23] Tran Viet Chuong, Ha Hai Nam *"A variable neighborhood search algorithm forsolving the Steiner minimal tree problem in sparse graphs"*, Context-Aware Systems and Applications, and Nature of Computation and Communication, EAI, Springer, pp.218-225, 2018.

[24] Thomas Pajor, Eduardo Uchoa, Renato F. Werneck, *"A robust and scalable algorithm for the Steiner problem in graphs"*, Springer, 2018.

[25] Thomas Bosman, *"A solution merging heuristic for the Steiner problem in graphs using tree decompositions"*, VU University Amsterdam, The Netherlands, pp.1-12, 2015.

[26] Thorsten Koch, Alexander Martin, *"Solving Steiner tree problems in graphs to optimality"*, Germany, pp.1-31, 1996.

[27] Trần Lê Thủy, *"Về bài toán Steiner"*, Viện Toán học, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 2014 (luận văn thạc sĩ).

[28] Vũ Đình Hòa, *"Bài toán Steiner"*, http://math.ac.vn.

[29] Wassim Jaziri (Edited). *"Local Search Techniques: Focus on Tabu Search"*. In-teh, Vienna, Austria, pp.1-201, 2008.

[30] Xinhui Wang, *"Exact algorithms for the Steiner tree problem"*, doctoral thesis, ISSN 1381-3617, 2008.

[31] Xiuzhen Cheng, Ding-zhu Du, *"Steiner trees in industry"*, Kluwer Academic Publishers, vol.5, pp.193-216, 2004.
[32] Xin-she Yang. *"Engineering optimization"*. Wiley, pp.21-137, 2010.
[33] S. E. Dreyfus, R. A. Wagner, *"The Steiner Problem in Graphs"*, Networks, Vol.1, pp.195-207, 1971.
[34] L. Kou, G. Markowsky, L. Berman, *"A Fast Algorithm for Steiner Trees"*, acta informatica, Vol.15, pp.141-145, 1981.

SƠ LƯỢC VỀ TÁC GIẢ

Trần Việt Chương

Sinh ngày: 01/12/1974

Nhận bằng Thạc sỹ Khoa học máy tính năm 2005 tại Trường Đại học Sư phạm I Hà Nội.

Hiện công tác tại Trung tâm CNTT và Truyền thông tỉnh Cà Mau, Sở Thông tin và Truyền thông tỉnh Cà Mau. Lĩnh vực nghiên cứu: Thuật toán tiến hóa và tối ưu.

Điện thoại: 0913 688 477

E-mail: chuongtv@camau.gov.vn

Phan Quốc Tấn

Sinh ngày: 12/10/1971

Nhận bằng Thạc sỹ Tin học tại Trường Đại học KHTN - ĐHQG TP.HCM năm 2002, Tiến sỹ chuyên ngành Khoa học máy tính tại Trường Đại học Bách Khoa Hà Nội năm 2015.

Hiện công tác tại Bộ môn Khoa học Máy tính, khoa Công nghệ Thông tin, Trường Đại học Sài Gòn. Lĩnh vực nghiên cứu: Thuật toán gần đúng.

Điện thoại: 0918 178 052

E-mail: quocpt@sgu.edu.vn

Hà Hải Nam

Sinh ngày: 10/02/1975

Nhận bằng Thạc sỹ Tin học năm 2004 tại Trường Đại học Bách Khoa Hà Nội, Tiến sỹ năm 2008 tại Vương quốc Anh. Được phong học hàm PGS năm 2014

Hiện công tác tại Viện Khoa học Kỹ thuật Bưu điện, Học viện Công nghệ Bưu chính Viễn thông. Lĩnh vực nghiên cứu: Hệ thống phân tán và tối ưu hóa.

Điện thoại: 091 66 34567

E-mail: namhh@ptit.edu.vn

Một phương pháp thiết kế ngữ nghĩa tính toán của các từ ngôn ngữ giải bài toán phân lớp dựa trên luật mờ

Nguyễn Đức Dư[1], Phạm Đình Phong[1], Phạm Đình Vũ[2], Nguyễn Đức Thảo[3]
[1]Khoa Công nghệ thông tin, Trường Đại học Giao thông vận tải
[2]Cục Công nghệ thông tin và Thống kê hải quan, Tổng cục Hải quan
[3]Viện Khoa học và Công nghệ quân sự
Tác giả liên hệ: Phạm Đình Phong, phongpd@utc.edu.vn
Ngày nhận bài: 20/01/2020, ngày sửa chữa: 17/06/2020
Định danh DOI: 10.32913/mic-ict-research-vn.v2020.n1.914

Tóm tắt: Thiết kế ngữ nghĩa tính toán của các từ ngôn ngữ trong cơ sở luật và biểu diễn cấu trúc của chúng đóng vai trò quan trọng trong việc nâng cao hiệu suất cũng như tính giải nghĩa được của hệ dựa trên luật mờ. Bài báo này trình bày phương pháp thiết kế ngữ nghĩa tính toán dựa trên tập mờ dạng hàm *S* được sinh bởi đại số gia tử mở rộng và được biểu diễn dưới dạng cấu trúc phân hoạch mờ đảm bảo tính giải nghĩa được của hệ phân lớp dựa trên luật mờ. Kết quả thực nghiệm với 23 tập dữ liệu chuẩn cho thấy hệ phân lớp với ngữ nghĩa tính toán dựa trên tập mờ dạng hàm *S* cho độ chính xác phân lớp tốt hơn so với ngữ nghĩa tính toán dựa trên tập mờ tam giác và hình thang cũng như chỉ ra tính hiệu quả của biểu diễn cấu trúc phân hoạch mờ đảm bảo tính giải nghĩa được của hệ phân lớp so với cấu trúc phân hoạch đã được đề xuất trước đó.

Từ khóa: *đại số gia tử, thứ tự ngữ nghĩa, hàm thuộc, hệ phân lớp dựa trên luật mờ.*

Title: **A Design Method of Computational Semantics of Linguistic Words for Fuzzy Rule-based Classifier**

Abstract: The design of computational semantics of linguistic terms in the fuzzy rule bases and structural representation of them play important roles in improving the performance and the interpretability of fuzzy rule-based systems. This paper presents a method of designing computational fuzzy sets-based semantics in form of *S*-shape membership function generated by the enlarged hedge algebras and represented as fuzzy partition structure to ensure the interpretability of the fuzzy rule-based classifiers. Experimental results over 23 real-world datasets have shown that the classifier with the fuzzy set-based computational semantics in form of *S*-shape membership function gives better classification accuracy than the ones previously proposed with triangular and trapezoidal fuzzy sets based semantics as well as shown the efficiency of the fuzzy partition structure representation which ensures the interpretability of the fuzzy rule-based classifiers in comparison with the existing ones.

Keywords: *hedge algebras, order-based semantics, membership function, fuzzy rule-based classifier.*

I. GIỚI THIỆU

Hệ phân lớp dựa trên luật mờ (Fuzzy Rule Based Classifier – FRBC) có nhiều ứng dụng trong lĩnh vực khai phá dữ liệu [1–4, 18–22] do mô hình phân lớp này có ưu điểm là dễ hiểu với người dùng và có thể sử dụng các tri thức dạng luật if-then được trích rút tự động từ dữ liệu như là tri thức của họ.

Trong [4, 5], Ishibuchi và Yamamoto đề xuất phương pháp trích rút hệ luật mờ tối giản cho FRBC từ cấu trúc phân hoạch mờ đa thể hạt được thiết kế sẵn bằng cách áp dụng một số kỹ thuật trong khai phá dữ liệu như độ tin cậy, độ hỗ trợ và trọng số luật kết hợp với thuật toán di truyền đa mục tiêu. Alcalá và các cộng sự đề xuất trong [1] một số phương pháp lựa chọn một đơn thể hạt tốt nhất trong số các thể hạt được thiết kế sẵn ban đầu do họ quan niệm rằng cấu trúc phân hoạch mờ đa thể hạt không giải nghĩa được. Sau đó thuật toán di truyền được áp dụng để lựa chọn hệ luật tối ưu đồng thời với tối ưu các tham số của các hàm thuộc. Một giản đồ tiến hóa đa mục tiêu nhanh và hiệu quả được Antonelli và các cộng sự đề xuất trong [2] có tên là PAES-RCS. Đây là một tiếp cận tiến hóa đa mục tiêu thực

hiện huấn luyện đồng thời cơ sở luật và cơ sở dữ liệu của FRBC. Trong pha đầu, tập luật mờ ứng cử viên được sinh từ các phân hoạch mờ được thiết kế sẵn bằng thuật toán C4.5. Sau đó, thuật toán tiến hóa đa mục tiêu được thực hiện để lựa chọn một tập luật mờ từ tập luật ứng cử viên đồng thời với lựa chọn các điều kiện của luật mờ cũng như hiệu chỉnh các tham số của hàm thuộc. Trong [5], Rey và các cộng sự đề xuất thêm một mục tiêu là tính thích hợp của luật (rule relevance) bên cạnh hai mục tiêu là tính chính xác (accuracy) và tính giải nghĩa được (interpretability) cho giải thuật tiến hóa đa mục tiêu lựa chọn hệ luật tối ưu cho hệ dựa trên luật mờ. Trong [18], Rudzinski đề xuất thuật toán tiến hóa đa mục tiêu thiết kế hệ phân lớp dựa trên luật mờ hướng tính giải nghĩa được. Trong quá trình huấn luyện, các tham số của các hàm thuộc và cấu trúc của cơ sở luật được tiến hóa đồng thời. Các độ đo về số tập mờ hoạt động và số biến đầu vào hoạt động (tức được sử dụng bởi ít nhất một luật) cùng với độ dài trung bình của luật được sử dụng để đánh giá tính giải nghĩa được của hệ phân lớp. Một mở rộng của giải thuật Chi nổi tiếng thiết kế hệ phân lớp dựa trên luật mờ phân tán cho phân lớp dữ liệu lớn bằng cách áp dụng khung làm việc dữ liệu lớn phổ biến Apache Hadoop, được đề xuất bởi Elkanoa và các cộng sự trong [19]. Trong [20] và [21] các tác giả đề xuất xây dựng các hệ phân lớp dựa trên luật mờ đặc thù áp dụng trong các lĩnh vực y tế và đánh giá rủi ro tín dụng. Một phương pháp thiết kế FRBC sử dụng giải thuật tiến hóa lượng tử đa dân số (Multi-population quantum evolutionary algorithm) với sự tái tạo lại luật mâu thuẫn được Zhang và các cộng sự đề xuất trong [22].

Như đã được trình bày ở trên, các phương pháp thiết kế FRBC trên cơ sở lý thuyết tập mờ [1–4, 18–22] trích rút các luật mờ từ các phân hoạch mờ được thiết kế sẵn trên miền giá trị của các thuộc tính sử dụng các tập mờ. Để nâng cao hiệu quả phân lớp, giá trị của các tham số hàm thuộc được hiệu chỉnh thích nghi bằng giải thuật tối ưu. Do không có cơ sở hình thức kết nối giữa ngữ nghĩa của các từ ngôn ngữ với các tập mờ nên ngữ nghĩa tính toán dựa trên tập mờ không phản ánh đúng ngữ nghĩa thực của các ngôn ngữ sau quá trình tối ưu và làm ảnh hưởng đến tính giải nghĩa được của hệ luật phân lớp.

Đại số gia tử (ĐSGT) [6–8] đã có những ứng dụng hiệu quả trong khai phá dữ liệu [9–12], điều khiển mờ [13], xử lý ảnh [14], lập lịch [15], ... ĐSGT khai thác tính thứ tự về ngữ nghĩa của các từ trong miền giá trị ngôn ngữ của biến ngôn ngữ để hình thành một cơ sở hình thức toán học cho việc liên kết ngữ nghĩa tính toán dựa trên tập mờ với ngữ nghĩa vốn có của các từ ngôn ngữ. Trên cơ sở đó, ĐSGT đã được ứng dụng hiệu quả để thiết kế tối ưu các từ ngôn ngữ cùng với ngữ nghĩa tính toán dựa trên tập mờ hình tam giác [9] và hình thang [10] cho các FRBC. Ngữ nghĩa tính toán dựa trên tập mờ hình thang có ưu điểm so với hình tam giác là biểu diễn được lõi ngữ nghĩa khoảng của các từ ngôn ngữ. Tuy nhiên, cả hai dạng tập mờ này đều có các cạnh được biểu diễn bởi các hàm tuyến tính có độ dốc lớn nên chưa thật mềm dẻo và gây mất mát thông tin lớn. Một phương pháp thiết kế ngữ nghĩa tính toán dựa trên tập mờ dạng hàm S và được sinh bởi ĐSGT mở rộng [10] cho các FRBC được trình bày trong bài báo này. Do hàm S là hàm phi tuyến nên phù hợp với sự biến thiên về ngữ nghĩa vốn có của các từ ngôn ngữ trong khi vẫn biểu diễn được lõi ngữ nghĩa khoảng của các từ ngôn ngữ.

Mặt khác, để đảm bảo tính giải nghĩa được của hệ dựa trên luật mờ được thiết kế theo tiếp cận ĐSGT, trong [11] các tác giả đã đưa ra bốn ràng buộc trên ngữ nghĩa tính toán của các từ ngôn ngữ. Các phương pháp thiết kế ngữ nghĩa tính toán dựa trên tập mờ cho FRBC của các công bố trong [9, 10] đều chưa thỏa tất cả bốn ràng buộc này. Cụ thể, một từ ngôn ngữ hx được sinh ra từ từ ngôn ngữ x bởi gia tử h có ngữ nghĩa cụ thể hơn x nhưng vẫn giữ nguyên ngữ nghĩa gốc của x. Ví dụ, từ ngôn ngữ "*rất trẻ*" được sinh ra từ từ ngôn ngữ "*trẻ*" bởi gia tử *rất* có ngữ nghĩa cụ thể hơn "*trẻ*" nhưng vẫn giữ được ngữ nghĩa gốc của "*trẻ*". Do đó, để thỏa ràng buộc thứ ba trong [11], trong biểu diễn cấu trúc phân hoạch mờ sử dụng các tập mờ thì độ hỗ trợ của tập mờ ứng với từ ngôn ngữ hx phải nằm trọn trong độ hỗ trợ của tập mờ ứng với từ ngôn ngữ x. Tuy nhiên, các thiết kế phân hoạch mờ trong [9, 10] không thỏa tính chất này. Bài báo này trình bày một phương pháp biểu diễn cấu trúc phân hoạch mờ sử dụng các tập mờ có dạng hàm S thỏa tất cả bốn ràng buộc trong [11], tức đảm bảo tính giải nghĩa được của hệ phân lớp dựa trên luật mờ.

Phần còn lại của bài báo được bố cục như sau: mục 2 trình bày tóm tắt ĐSGT mở rộng và hệ phân lớp dựa trên luật mờ với ngữ nghĩa dựa trên tập mờ dạng hàm S; mục 3 trình bày kết quả thực nghiệm và thảo luận; một số kết luận rút ra trong mục 4.

II. NỘI DUNG NGHIÊN CỨU

1. Một số khái niệm cơ bản về đại số gia tử mở rộng

ĐSGT mở rộng [10] được xây dựng bằng việc bổ sung một gia tử nhân tạo h_0 nhằm mô hình hóa lõi ngữ nghĩa của các từ ngôn ngữ.

Một cấu trúc $\mathcal{AX}^{en} = (X_{en}, G, C, H_{en}, \leq)$ được gọi là ĐSGT mở rộng (ĐSGTMR) của ĐSGT tuyến tính và sinh tự do $\mathcal{AX}$ nếu thỏa các tiên đề bổ sung sau:

(A1) $h_0x \notin H(G) = \{\sigma c \mid c \in G\}$ và $hh_0x = h_0x$ luôn là điểm bất động.

(A2) $h_px \geq x \Rightarrow h_{-q}x \leq \ldots \leq h_{-1}x \leq h_0x \leq h_1x \leq \ldots \leq h_px$

$h_px \leq x \Rightarrow h_px \leq \ldots \leq h_1x \leq h_0x \leq h_{-1}x \leq \ldots \leq h_{-q}x.$

Một hàm fm: $X_{en} \rightarrow [0, 1]$ được gọi là độ đo tính mờ của ĐSGTMR $\mathcal{AX}^{en}$ nếu nó thỏa các tính chất sau:

(F1): $fm(\mathbf{0}) + fm(c^-) + fm(W) + fm(c^+) + fm(\mathbf{1}) = 1$;

(F2): $\sum_{h \in H_{en}} fm(hx) = fm(x)$ với $\forall x \in H(G)$;

(F3): $\forall x, y \in H(G), \forall h \in H_{en}$ tỷ lệ $fm(hx)/fm(hy) = fm(x)/fm(y)$ không phụ thuộc vào bất kỳ từ ngôn ngữ nào trong X_{en} được gọi là độ đo tính mờ của gia tử h và được ký hiệu là $\mu(h)$.

Độ đo tính mờ của một từ ngôn ngữ của ĐSGTMR X_{en} thỏa các tính chất sau:

(1) $\sum_{x \in X_{(k)}} fm(x) = 1, k > 0$. Với $k = 1$ thì $fm(\mathbf{0}) + fm(c^-) + fm(W) + fm(c^+) + fm(\mathbf{1}) = 1$;

(2) $\sum_{h \in H_{en}} \mu(h) = 1$

(3) $fm(hx) = \mu(h) fm(x)$, với $\forall h \in H_{en}$, $\forall x \in H(\{c^-, c^+\})$ và $hx \neq x$;

(4) $fm(x) = \mu(h_n)...\mu(h_1) fm(c)$, trong đó $x = h_n...h_1 c$, $c \in \{c^-, c^+\}$, là biểu diễn chính tắc của $x \in X_{en}$.

Cho độ đo tính mờ $fm : X_{en} \to [0, 1]$ của một ĐSGTMR $\mathcal{AX}^{en}$ của biến một ngữ $\mathcal{X}$ và mỗi từ $x \in X_{en}$ được liên kết với một khoảng $\Im(x) \subseteq [0, 1]$. Các khoảng này được gọi là các khoảng tính mờ ứng với các từ của $\mathcal{X}$ nếu thỏa các điều kiện sau:

(FI1): $|\Im(x)| = fm(x)$ với $\forall x \in X_{en}$ và $|\Im(x)|$ biểu thị độ dài của khoảng $\Im(x)$;

(FI2): Tập $\{\Im(hx) | x \in X_{en}\}$ tạo thành một phân hoạch của $\Im(x)$ và có thứ tự tương đồng với thứ tự của các từ ngôn ngữ liên kết với chúng.

Khoảng tính mờ mức k của x được ký hiệu là $\Im_k(x)$. Quy ước rằng các khoảng tính mờ là mở phải và đóng trái, khoảng tính mờ của hằng tử $\mathbf{1}$ là đóng cả hai phía.

Ánh xạ ngữ nghĩa định lượng khoảng $f(x)$ của từ ngôn ngữ x được xác định là hàm $f(x) = \Im(h_0 x), x \in X_{en}$ và khẳng định này đã được chứng minh trong [10].

2. Thiết kế FRBC với ngữ nghĩa tính toán dựa trên tập mờ dạng hàm S

Bài toán thiết kế hệ phân lớp dựa trên luật mờ $\mathcal{P}$ được định nghĩa như sau: Một tập $\boldsymbol{P} = \{(\boldsymbol{d}_p, C_p) | \boldsymbol{d}_p \in \boldsymbol{D}, C_p \in \boldsymbol{C}, p = 1, \ldots, m\}$ gồm m mẫu dữ liệu, trong đó $\boldsymbol{d}_p = [d_{p,1}, d_{p,2}, ..., d_{p,n}]$ là dòng thứ p^{th}, $\boldsymbol{C} = \{C_s | s = 1, \ldots, M\}$ là tập gồm M nhãn lớp, n là số thuộc tính.

Hệ cơ sở luật cho bài toán phân lớp được sử dụng trong bài báo này là tập luật có trong số dưới dạng:

$$\text{Luật } R_q : \textbf{If } \mathcal{X}_1 \text{ is } A_{q,1} \textbf{ and } ... \textbf{ and } \mathcal{X}_n \text{ is } A_{q,n} \textbf{ then} C_q \, with \, CF_q, for \; q = 1, \ldots, N \quad (1)$$

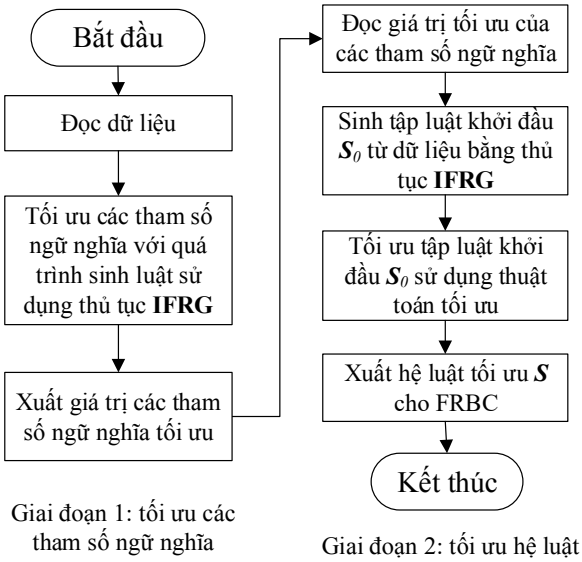
trong đó $\chi = \{\mathcal{X}_j, j = 1, \ldots, n\}$ là tập n biến ngôn ngữ ứng với n thuộc tính của tập dữ liệu $\boldsymbol{P}$; $A_{q,j}$ là các giá trị ngôn ngữ của thuộc tính thứ $j, F_j; C_q$ là nhãn lớp và CF_q là trọng số của luật R_q. Luật R_q được viết gọn lại như sau:

$$\boldsymbol{A}_q \Rightarrow C_q \text{ with } CF_q, \text{ với } q = 1, \ldots, N \quad (2)$$

trong đó $\boldsymbol{A}_q$ là tiền đề của luật thứ q.

Giải bài toán $\mathcal{P}$ là trích xuất từ tập dữ liệu $\mathbf{P}$ một tập luật $\mathbf{S}$ có dạng (1) nhỏ gọn, dễ hiểu với người dùng và có độ chính xác phân lớp cao. Phương pháp thiết kế hệ phân lớp dựa trên luật mờ theo tiếp cận ĐSGT gồm hai bước (xem Hình 1):

1) Thiết kế tối ưu các từ ngôn ngữ cùng với ngữ nghĩa tính toán dựa trên tập mờ của chúng sử dụng giải thuật tối ưu. Sau bước này ta thu được bộ tham số ngữ nghĩa tối ưu.
2) Trích xuất từ tập dữ liệu huấn luyện tập luật tối ưu cho hệ phân lớp trên cơ sở thỏa hiệp giữa tính dễ hiểu và độ chính xác của hệ phân lớp sử dụng giải thuật tối ưu.



Hình 1. Phương pháp hai bước thiết kế FRBC

ĐSGTMR là cung cấp một cơ sở hình thức cho phép ngữ nghĩa định tính xác định giá trị ngữ nghĩa định lượng khoảng của các từ ngôn ngữ, và trên cơ sở đó ngữ nghĩa dựa trên tập mờ có lõi là một khoảng của chúng được xây dựng. Trong bài báo này chúng tôi sử dụng ĐSGTMR để sinh ngữ nghĩa dựa trên tập mờ có dạng hàm S có lõi là một khoảng cho hệ phân lớp dựa trên luật mờ.

Mỗi ĐSGT $\mathcal{AX}_j^{en}$ được liên kết với một thuộc tính thứ j của tập dữ liệu cảm sinh các từ ngôn ngữ $X_{j,(k_j)}$ có độ dài lớn nhất k_j theo thứ tự ngữ nghĩa của chúng. Vì ngữ nghĩa định lượng khoảng $f(x_{j,i}) = \Im(h_0 x_{j,i}) \subseteq \Im(x_{j,i})$ biểu thị lõi ngữ nghĩa của từ ngôn ngữ $x_{j,i}$ nên được dùng để biểu diễn đỉnh của tập mờ dạng hàm S ứng với từ $x_{j,i}$. Các giá

trị trong khoảng đỉnh của tập mờ phù hợp với ngữ nghĩa định tính của từ nhất nên có giá trị là 1.

Ký kiệu $\mathcal{L}(\bullet)$ và $\mathcal{R}(\bullet)$ lần lượt là điểm mút trái và mút phải của một khoảng bất kỳ. Giả sử đặt $a = \mathcal{R}(f(x_{j,i-1}))$, $c = \mathcal{L}(f(x_{j,i}))$, $d = \mathcal{R}(f(x_{j,i}))$, $g = \mathcal{L}(f(x_{j,i+1}))$, khi đó $b = a + (c-a)/4, e = d + (g-e)/4$ và v là một điểm dữ liệu. Ta có hàm biểu diễn độ thuộc của v vào nửa trái của hàm S, S_{left} như sau:

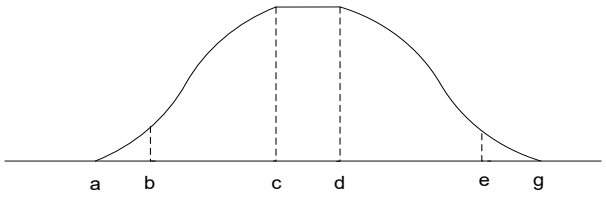
$$S_{left} = \begin{cases} 0, & 0 \le v \le a \\ \frac{(v-a)^2}{(b-a)(c-a)}, & a \le v \le b \\ 1 - \frac{(v-c)^2}{(c-b)(c-a)}, & b \le v \le c \\ 1, & v \ge c \end{cases}$$

và hàm biểu diễn độ thuộc của v vào nửa phải của hàm S, S_{right} như sau:

$$S_{right} = \begin{cases} 1, & 0 \le v \le d \\ 1 - \frac{(v-d)^2}{(d-e)(d-g)}, & d \le v \le e \\ \frac{(v-g)^2}{(e-d)(g-d)}, & e \le v \le g \\ 0, & v \ge g \end{cases}$$

Tập mờ dạng hàm S được biểu diễn như Hình 2.

Trong bài báo này, tập mờ dạng hàm S được sử dụng để phân hoạch miền giá trị thuộc tính của tập dữ liệu dưới dạng cấu trúc đa thể hạt được đề xuất trong [10], được gọi là phân hoạch k_1 và dưới dạng cấu trúc đa thể hạt được đề xuất trong [11] với mức $k = 1$ được tách thành hai mức 0 và 1, được gọi là phân hoạch k_0.



Hình 2. Biểu diễn tập mờ dạng hàm S

Trong cấu trúc phân hoạch k_1 mỗi thể hạt được phân hoạch bởi các tập mờ ứng với các từ ngôn ngữ có độ dài bằng nhau và hai phần tử 0 và 1, và theo thứ tự ngữ nghĩa của các từ ngôn ngữ tương ứng. Cấu trúc phân hoạch k_0 khác với k_1 là mức $k = 1$ gồm các từ ngôn ngữ có độ dài bằng 1 được tách thành hai thể hạt: thể hạt thứ nhất (mức $k = 0$) gồm các hằng tử $\mathbf{0}_0$, W và $\mathbf{1}_0$, và thể hạt thứ hai (mức $k = 1$) gồm 4 từ ngôn ngữ $\mathbf{0}_1$, c^-, c^+ và $\mathbf{1}_1$. Với cách thiết kế này, độ hỗ trợ của tập mờ ứng với từ ngôn ngữ x hoàn toàn chứa độ hỗ trợ của từ ngôn ngữ hx và trong [11] đã chứng minh phân hoạch k_0 đảm bảo tính giải nghĩa được của hệ dựa trên luật mờ.

Với các từ ngôn ngữ không phải là các hằng tử $\mathbf{0}$ và $\mathbf{1}$, giá trị của a là giá trị đầu mút phải của giá trị định lượng khoảng của từ gần nhất bên trái có cùng độ dài và giá trị g là đầu mút trái của giá trị định lượng khoảng của từ gần nhất bên phải có cùng độ dài. Ví dụ, Hình 3 biểu diễn cấu trúc phân hoạch k_1 và Hình 4 biểu diễn cấu trúc phân hoạch k_0, sử dụng các tập mờ dạng hàm S với độ dài tối đa của các từ ngôn ngữ $k_j = 2$. Trong đó, tập mờ ứng với từ Lc^+ có mút trái $a = \mathcal{R}(f(Lc^-))$ và mút phải $g = \mathcal{L}(f(Vc^+))$, tương tự với các tập mờ khác.

Với giá trị cụ thể của các tham số ngữ nghĩa bao gồm $fm(c^-)$, $fm(W_j)$, $fm(\mathbf{0}_j)$, $fm(\mathbf{1}_j)$, $\mu(h_{j,i})$, $\mu(h_{j,0})$ là độ đo tính mờ tương ứng của c_j^- , W_j, $\mathbf{0}_j$, $\mathbf{1}_j$, $h_{j,i}$, $h_{j,0}$ và với giá trị cụ thể của k_j, các khoảng tính mờ $\Im_k(x_{j,i})$, $x_{j,i} \in X_{j,k}, k \le k_j$ và các ngữ nghĩa định lượng khoảng $f(x_{j,i})$ được tính toán. Các khoảng tính mờ $\Im_{k_j}(x_{j,i})$ tạo thành phân hoạch mức k_j trên miền giá trị của thuộc tính j. Có duy nhất một khoảng tính mờ trong số các khoảng tính mờ $\Im_{k_j}(x_{j,i})$ chứa điểm dữ liệu $d_{p,j}$ của mẫu dữ liệu d_p. Tất cả các khoảng tính mờ mức k_j chứa $d_{p,j}$ $(0 \le j \le n)$ tạo thành một siêu hộp $\mathcal{H}_p$ và chỉ sinh các luật mờ từ các siêu hộp loại này. Luật mờ cơ sở có độ dài n được sinh từ $\mathcal{H}_p$ với nhãn lớp C_p của mẫu dữ liệu d_p có dạng sau:

if $\boldsymbol{X}_1$ is $x_{1,i(1)}$ **and** ... **and** $\boldsymbol{X}_n$ is $x_{n,i(n)}$ **then** C_p (R_b)

Các luật mờ thứ cấp có độ dài $L \le n$ thu được bằng cách bỏ bớt $n - L$ thuộc tính có dạng sau:

if $\boldsymbol{X}_{j1}$ is $x_{j1,i(j1)}$ **and**...**and** $\boldsymbol{X}_{jt}$ is $x_{jt,i(jt)}$**then** C_q (R_{snd})

trong đó $1 \le j_1 \le ... \le j_t \le n$. Nhãn lớp C_q của luật R_q được xác định bởi độ tin cậy $c(\boldsymbol{A}_q \Rightarrow C_h)$ [3, 4] của R_q:

$$C_q = \text{argmax}(c(\boldsymbol{A}_q \Rightarrow C_h) \mid h = 1, \ldots, M) \qquad (3)$$

Độ tin cậy của luật mờ được tính như sau:

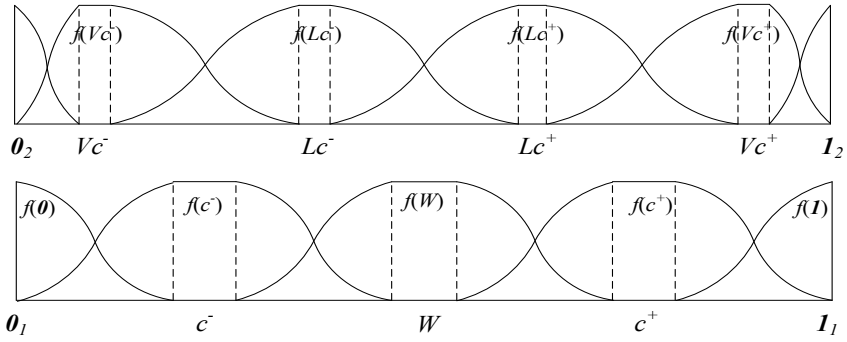
$$c(\boldsymbol{A}_q \Rightarrow C_h) = \sum_{d_p \in C_h} \mu_{\boldsymbol{A}_q}(d_p) / \sum_{p=1}^{m} \mu_{\boldsymbol{A}_q}(d_p) \qquad (4)$$

trong đó $\mu_{\boldsymbol{A}_q}(d_p)$ là độ đốt cháy của mẫu dữ liệu d_p đối với tiền đề luật của R_q và thường được tính bằng biểu thức toán tử nhân theo công thức sau:

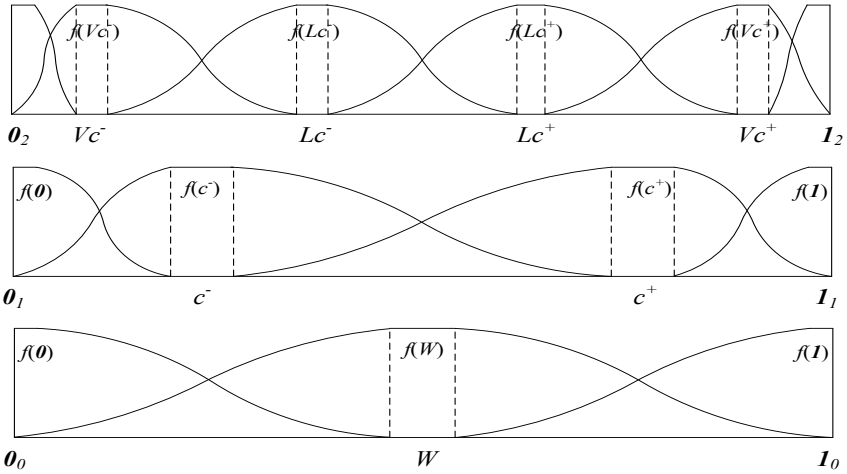
$$\mu_{\boldsymbol{A}_q}(d_p) = \prod_{j=1}^{n} \mu_{q,j}(d_{p,j}). \qquad (5)$$

với $\mu_{q,j}(d_{p,j})$ là độ thuộc của điểm dữ liệu $d_{p,j}$ vào tập mờ $A_{q,j}$.

Tập luật ứng viên thu được sau khi sàng lọc các luật không nhất quán bằng độ hỗ trợ. Tiếp theo, một tiêu chuẩn sàng được sử dụng để chọn ra tập luật khởi đầu $\boldsymbol{S}_0$ gồm $NR_0 = NB_0 \times M$ luật với M là số nhãn lớp và NB_0 là số luật dành cho mỗi lớp. Tiêu chuẩn sàng được chọn có thể là độ tin cậy c (công thức (4)), độ hỗ trợ s hoặc tích $c \times s$. Độ hỗ trợ được tính theo công thức sau [3]:



Hình 3. Cấu trúc phân hoạch k_1 với tập mờ dạng hàm S và $k_j = 2$



Hình 4. Cấu trúc phân hoạch k_0 với tập mờ dạng hàm S và $k_j = 2$

$$s(\boldsymbol{A}_q \Rightarrow C_h) = \sum_{d_p \in C_h} \mu_{\boldsymbol{A}_q}(d_p)/m. \quad (6)$$

Mỗi luật được gán một trọng số để nâng cao độ chính xác phân lớp. Trong bài báo này, trọng số luật được tính theo công thức [3]:

$$CF_q = c\left(\boldsymbol{A}_q \Rightarrow C_q\right) - c_{q,2nd}, \quad (7)$$

trong đó $c_{q,2nd}$ là độ tin cậy lớn nhất của các luật có cùng tiền đề A_q nhưng khác kết luận C_q:

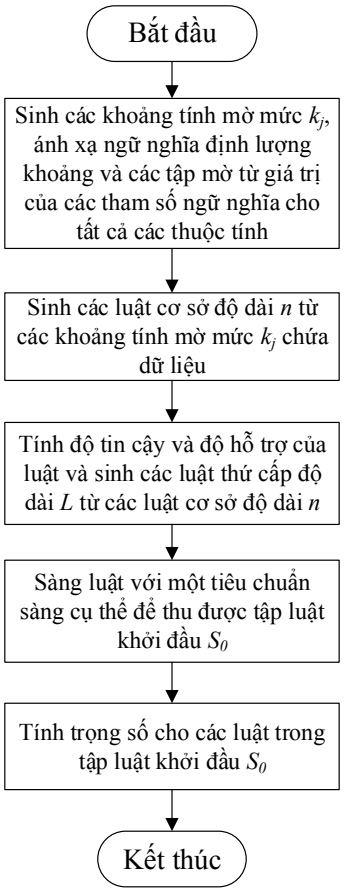
$$c_{q,2nd} = \max(c(\boldsymbol{A}_q \Rightarrow Class\ h) \mid h = 1, \ldots, M;\ h \neq C_q) \quad (8)$$

Quá trình sinh luật trên là thủ tục sinh tập luật khởi đầu ***IFRG***(Π, ***P***, NR_0, L) [9], trong đó Π là tập giá trị của các tham số ngữ nghĩa và L là số tiền đề tối đa của mỗi luật. Thủ tục này được trực quan hóa như được thể hiện trong Hình 5. Độ phức tạp của thủ tục sinh tập luật khởi đầu **IFRG** là đa thức đối với số mẫu và số thuộc tính của tập dữ liệu ***D*** và đã được chứng minh trong [9].

Mỗi loại dữ liệu có sự phân bố dữ liệu khác nhau cần bộ tham số ngữ nghĩa phù hợp để nâng cao hiệu suất phân lớp. Do đó, một thuật toán tối ưu được áp dụng để tìm bộ tham số ngữ nghĩa tối ưu và chúng được sử dụng để sinh tập luật khởi đầu làm đầu vào cho thủ tục lựa chọn tập luật nhỏ gọn và dễ hiểu cho hệ phân lớp trên cơ sở thỏa hiệp giữa độ chính xác và độ phức tạp của hệ phân lớp.

III. KẾT QUẢ THỰC NGHIỆM VÀ THẢO LUẬN

Mục này trình bày các kết quả thực nghiệm của các hệ phân lớp dựa trên luật mờ sử dụng cấu trúc phân hoạch k_0 và k_1 với ngữ nghĩa tính toán dựa trên tập mờ có dạng hàm S của các từ ngôn ngữ và so sánh đánh giá với các hệ phân lớp khác để minh chứng tính hiệu quả của các hệ phân lớp được đề xuất.



Hình 5. Lưu đồ thủ tục sinh tập luật khởi đầu

1. Cài đặt thực nghiệm

Các thực nghiệm được được cài đặt bằng ngôn ngữ C# chạy trên Windows 7. Các tập dữ liệu thực nghiệm được lấy từ nguồn KEEL-Dataset tại địa chỉ http://sci2s.ugr.es/keel/datasets.php. Phương pháp kiểm tra chéo 10 nhóm được áp dụng để huấn luyện và kiểm tra. Để đảm bảo sự khác biệt của các kết quả thực nghiệm của các hệ phân lớp được so sánh là có ý nghĩa, phương pháp kiểm định giả thuyết thống kê Wilcoxon [16] được sử dụng để kiểm tra giả thuyết H_0 (null hypothesis) có độ tin cậy là 90% ($\alpha = 0,1$) với giả định rằng các kết quả của các phương pháp được so sánh là tương đương nhau.

Nhằm giảm không gian tìm kiếm trong quá trình huấn luyện, các ràng buộc về giá trị của các tham số ngữ nghĩa được áp dụng như sau: số gia tử âm và số gia tử dương là 1, gia tử âm là "*Less*" (L) và gia tử dương là "*Very*" (V); $1 \leq k_j \leq 3$; $0,2 \leq \left\{fm\left(c_j^-\right), fm\left(c_j^+\right)\right\} \leq 0,7$; $0,00001 \leq \{fm(\mathbf{0}_j), fm(\mathbf{1}_j)\} \leq 0,01$; $0,0001 \leq fm(W_j) \leq 0,2$; $fm\left(0_j\right) + fm\left(c_j^-\right) + fm\left(W_j\right) + fm\left(c_j^+\right) + fm\left(1_j\right) = 1$; $0,2 \leq \{\mu(L_j), \mu(V_j)\} \leq 0,7$; $0,01 \leq \mu(h_{0,j}) \leq 0,5$; and $\mu(L_j) + \mu(V_j) + \mu(h_{0,j}) = 1$.

Thuật toán tối ưu bầy đàn đa mục tiêu (PSO) [17] được sử dụng cho các bài toán tối ưu. Trong tối ưu các tham số ngữ nghĩa: số thế hệ là 250; số cá thể mỗi thế hệ là 600; hệ số Inertia là 0.4; hệ số nhận thức cá nhân là 0,2; hệ số nhận thức xã hội là 0,2; số luật khởi tạo bằng số thuộc tính; độ dài tối đa của luật là 1. Trong tối ưu hệ luật: số thế hệ là 1500; số luật khởi tạo là $|\boldsymbol{S}_0| = 300 \times$ số lớp; độ dài tối đa của luật là 3. Phương pháp lập luận phân lớp được sử dụng trong tất cả các thực nghiệm là *single winner rule* [3, 4], tiêu chuẩn sàng luật là $c \times s$ và trọng số luật được tính toán theo công (7).

2. Kết quả thực nghiệm

Ký hiệu hệ phân lớp với ngữ nghĩa tính toán dựa trên tập mờ dạng hàm S với phân hoạch k_0 [11] và phân hoạch k_1 [10] tương ứng là **FRBC_S_k0** và **FRBC_S**, hệ phân lớp với ngữ nghĩa tính toán dựa trên tập mờ hình thang với phân hoạch k_0 và phân hoạch k_1 tương ứng là **FRBC_TRA_k0** và **FRBC_TRA**, hệ phân lớp với ngữ nghĩa tính toán dựa trên tập mờ tam giác trong [9] là **FRBC_TRI**. Bảng I thể hiện các kết quả thực nghiệm và so sánh giữa các hệ phân lớp nêu trên, trong đó chữ đậm thể hiện kết quả tốt hơn so với các hệ phân lớp còn lại. Ký hiệu #R×C là độ phức tạp của hệ phân lớp (tích của số luật trung bình và số điều kiện trung bình của các luật), P_{te} là độ chính xác phân lớp trung bình trên tập kiểm tra.

Các kết quả thực nghiệm trong Bảng I cho thấy, hệ phân lớp **FRBC_S_k0** có độ chính xác phân lớp trên tập kiểm tra cao hơn so với các hệ phân lớp **FRBC_S**, **FRBC_TRA_k0**, **FRBC_TRA** và **FRBC_TRI** tương ứng đối với 15, 18, 17 và 20 trong số 23 tập dữ liệu được thực nghiệm. So sánh dựa trên độ chính xác phân lớp trung bình của 23 tập dữ liệu được thực nghiệm, hệ phân lớp **FRBC_S_k0** có độ chính xác phân lớp trung bình là 83,04%, cao nhất so với các hệ phân lớp còn lại. So sánh dựa trên độ phức tạp của hệ phân lớp, các hệ phân lớp không có sự chênh lệch nhiều. Ngoài ra, hệ phân lớp **FRBC_S** có độ chính xác phân lớp trung bình là 82,79%, cao hơn so với hệ phân lớp **FRBC_TRA** và **FRBC_TRI** lần lượt có độ chính xác phân lớp trung bình là 82,67% và 81,92%. Các kết quả kiểm định giả thuyết thống kê Wilcoxon [16] với độ tin cậy 90% ($\alpha = 0,1$) sử dụng dữ liệu trong Bảng I với giả thiết độ chính xác phân lớp và độ phức tạp tương ứng của hai hệ phân lớp là tương đương nhau (Giả thuyết H_0) được thể hiện trong Bảng II và Bảng III. Các giá trị *Exact p-value* trong Bảng II đều nhỏ hơn $\alpha = 0,1$ cho biết rằng giả thuyết tương đương H_0 về độ chính xác phân lớp giữa các hệ phân lớp được so sánh bị bác bỏ. Điều này có nghĩa là hệ phân lớp **FRBC_S_k0** có độ chính xác phân lớp cao hơn so với các hệ phân lớp **FRBC_S**, **FRBC_TRA_k0**, **FRBC_TRA** và **FRBC_TRI**; hệ phân lớp **FRBC_S** có độ chính xác phân lớp cao hơn so

Bảng I
KẾT QUẢ THỰC NGHIỆM CỦA CÁC HỆ PHÂN LỚP **FRBC_S_K0**, **FRBC_S**, **FRBC_TRA_K0**, **FRBC_TRA** VÀ **FRBC_TRI**

STT	Tập dữ liệu	FRBC_S_k0		FRBC_S		FRBC_TRA_k0		FRBC_TRA		FRBC_TRI	
		#R×C	P_{te}	#R×C	P_{te}	#R×C	P_{te}	#R×C	P_{te}	#R×C	P_{te}
1	Appendicitis	23,30	**88,73**	17,35	88,48	19,90	88,64	**16,77**	88,15	21,32	87,55
2	Australian	46,23	**87,54**	**35,93**	87,25	46,16	87,49	46,50	87,15	36,20	86,38
3	Bands	59,40	73,00	55,80	73,40	61,80	72,95	58,20	**73,46**	**52,20**	72,80
4	Bupa	**177,72**	72,03	221,65	72,19	186,05	71,97	181,19	**72,38**	187,20	68,09
5	Cleveland	509,54	61,73	**433,16**	61,86	703,17	61,14	468,13	**62,39**	657,43	62,19
6	Dermatology	240,11	**96,26**	254,98	94,50	216,50	96,17	**182,84**	94,40	198,05	96,07
7	Glass	467,18	**72,97**	364,08	72,30	400,20	72,32	474,29	72,24	**343,60**	72,09
8	Haberman	12,00	77,42	16,00	**77,43**	12,00	77,41	10,80	77,40	**10,20**	75,76
9	Hayes-roth	117,14	**85,21**	136,65	84,36	128,44	84,58	**114,66**	84,17	122,27	84,17
10	Heart	117,24	84,94	**95,25**	84,69	124,75	**85,43**	123,29	84,57	122,72	84,44
11	Hepatitis	26,10	**91,22**	36,63	89,99	25,95	**91,22**	**25,53**	89,28	26,16	88,44
12	Ionosphere	98,81	**92,32**	92,83	91,65	96,91	92,22	**88,03**	91,56	90,33	90,22
13	Iris	**16,52**	**98,00**	17,76	97,33	21,73	97,78	30,37	97,33	26,29	96,00
14	Mammogr.	77,87	**84,36**	76,84	84,25	**49,67**	84,33	73,84	84,2	92,25	84,20
15	Newthyroid	44,55	**96,59**	49,98	95,84	41,50	96,00	**39,82**	95,67	45,18	94,42
16	Pima	62,11	76,45	**47,55**	**77.17**	57,70	77,09	56,12	77,01	60,89	76,18
17	Saheart	95,24	**71,07**	68,13	70,42	89,79	70,71	**59,28**	70,05	86,75	69,33
18	Sonar	59,29	77,98	62,32	**79,43**	53,86	77,95	**49,31**	78,61	79,76	76,80
19	Tae	**163,80**	61,22	176,48	**61,44**	176,06	61,43	210,70	61,00	261,00	59,47
20	Vehicle	177,29	68,48	207,91	**68,88**	**163,80**	68,41	195,07	68,20	242,79	67,62
21	Wdbc	27,88	96,19	35,85	95,90	28,00	96,72	**25,04**	96,78	37,35	**96,96**
22	Wine	36,73	**98,87**	46,79	98,51	36,37	98,50	40,39	98,49	**35,82**	98,30
23	Wisconsin	91,27	97,34	73,66	96,80	79,82	97,05	**69,81**	**96,95**	74,36	96,74
Trung bình		**119,45**	**83,04**	**114,07**	**82,79**	**122,61**	**82,94**	**114,78**	**82,67**	**126,53**	**81,92**

Bảng II
SO SÁNH ĐỘ CHÍNH XÁC GIỮA CÁC HỆ PHÂN LỚP **FRBC_S_K0**, **FRBC_S**, **FRBC_TRA_K0**, **FRBC_TRA** VÀ **FRBC_TRI** BẰNG PHƯƠNG PHÁP KIỂM ĐỊNH WILCOXON VỚI $\alpha = 0,1$

So sánh (α = 0,1)	R^+	R^-	Exact P-value	Giả thuyết H_0
FRBC_S_k0 vs **FRBC_S**	196,0	80,0	0,0802	Bị bác bỏ
FRBC_S_k0 vs **FRBC_TRA_k0**	188,0	65,0	0,04616	Bị bác bỏ
FRBC_S_k0 vs **FRBC_TRA**	208,0	68,0	0,03266	Bị bác bỏ
FRBC_S vs **FRBC_TRA**	188,5	64,5	0,04433	Bị bác bỏ
FRBC_S vs **FRBC_TRI**	240,0	36,0	0,0011184	Bị bác bỏ

Bảng III
SO SÁNH ĐỘ PHỨC TẠP GIỮA CÁC HỆ PHÂN LỚP **FRBC_S_K0**, **FRBC_S**, **FRBC_TRA_K0**, **FRBC_TRA** VÀ **FRBC_TRI** BẰNG PHƯƠNG PHÁP KIỂM ĐỊNH WILCOXON VỚI $\alpha = 0,1$.

So sánh (α = 0,1)	R^+	R^-	Exact P-value	Giả thuyết H_0
FRBC_S_k0 vs **FRBC_S**	133,0	143,0	$\geq 0,2$	Không bị bác bỏ
FRBC_S_k0 vs **FRBC_TRA_k0**	126,0	150,0	$\geq 0,2$	Không bị bác bỏ
FRBC_S_k0 vs **FRBC_TRA**	99,0	177,0	$\geq 0,2$	Không bị bác bỏ
FRBC_S vs **FRBC_TRA**	115,0	161,0	$\geq 0,2$	Không bị bác bỏ
FRBC_S vs **FRBC_TRI**	161,0	115,0	$\geq 0,2$	Không bị bác bỏ

với hai hệ phân lớp sử dụng cùng cấu trúc phân hoạch k_1 là **FRBC_TRA** và **FRBC_TRI**. Các giá trị *Exact p-value* trong Bảng III đều lớn hơn $\alpha = 0,1$ nên giả thuyết tương đương H_0 về độ phức tạp của các hệ phân lớp không bị bác bỏ. Do đó, ta có thể khẳng định rằng, với cùng một cách biểu diễn phân hoạch mờ thì các hệ phân lớp dựa trên luật mờ với ngữ nghĩa tính toán dựa trên tập mờ của các từ ngôn ngữ có dạng hàm S được sinh bởi ĐSGT mở rộng cho độ chính xác phân lớp cao hơn so với dạng hình tam giác và hình thang do hàm S biểu diễn sự biến thiên về ngữ nghĩa tốt hơn. Ngoài ra, cấu trúc phân hoạch k_0 cho hiệu suất phân lớp tốt hơn cấu trúc phân hoạch k_1 đồng thời đảm bảo tính giải nghĩa được của hệ phân lớp như đã được chứng minh trong [11].

Nhằm thể hiện tính hiệu quả của hệ phân lớp với ngữ nghĩa tính toán dựa trên tập mờ dạng hàm S được sinh bởi ĐSGT mở rộng được đề xuất so với tiếp cận lý thuyết tập mờ, các kết quả thực nghiệm của hệ phân lớp **FRBC_S** được so sánh với các kết quả của hai hệ phân lớp **PAES-RCS** và **FURIA** [2]. Kết quả so sánh trong Bảng IV cho thấy, hệ phân lớp **FRBC_S** cho độ chính xác phân lớp trên

Bảng IV
KẾT QUẢ THỰC NGHIỆM CỦA HỆ PHÂN LỚP **FRBC_S**, **PAES-RCS** VÀ **FURIA**

STT	Tập dữ liệu	FRBC_S		PAES-RCS		$\neq P_{te}$	$\neq R{\times}C$	FURIA		$\neq P_{te}$	$\neq R{\times}C$
		$\#R{\times}C$	P_{te}	$\#R{\times}C$	P_{te}			$\#R{\times}C$	P_{te}		
1	Appendicitis	17,35	88,48	35,28	85,09	3,39	-17,93	19,00	85,18	3,30	-1,65
2	Australian	35,93	87,25	329,64	85,80	1,45	-293,71	89,60	85,22	2,03	-53,67
3	Bands	55,80	73,40	756,00	67,56	5,84	-700,20	535,15	64,65	8,75	-479,35
4	Bupa	221,65	72,19	256,20	68,67	3,52	-34,55	324,12	69,02	3,17	-102,47
5	Cleveland	433,16	61,86	1140,00	59,06	2,80	-706,84	134,67	56,20	5,66	298,49
6	Dermatology	254,98	94,50	389,40	95,43	-0,93	-134,42	303,88	95,24	-0,74	-48,90
7	Glass	364,08	72,30	487,90	72,13	0,17	-123,82	474,81	72,41	-0,11	-110,73
8	Haberman	16,00	77,43	202,41	72,65	4,78	-186,41	22,04	75,44	1,99	-6,04
9	Hayes-roth	136,65	84,36	120,00	84,03	0,33	16,65	188,10	83,13	1,23	-51,45
10	Heart	95,25	84,69	300,30	83,21	1,48	-205,05	193,64	80,00	4,69	-98,39
11	Hepatitis	36,63	89,99	300,30	83,21	6,78	-263,67	52,38	84,52	5,47	-15,75
12	Ionosphere	92,83	91,65	670,63	90,40	1,25	-577,80	372,68	91,75	-0,10	-279,85
13	Iris	17,76	97,33	69,84	95,33	2,00	-52,08	31,95	94,66	2,67	-14,19
14	Mammogr.	76,84	84,25	132,54	83,37	0,88	-55,70	16,83	83,89	0,36	60,01
15	Newthyroid	49,98	95,84	97,75	95,35	0,49	-47,77	100,82	96,30	-0,46	-50,84
16	Pima	47,55	77.17	270,64	74,66	2,51	-223,09	127,50	74,62	2,55	-79,95
17	Saheart	68,13	70,42	525,21	70,92	-0,50	-457,08	50,88	69,69	0,73	17,25
18	Sonar	62,32	79,43	524,60	77,00	2,43	-462,28	309,96	82,14	-2,71	-247,64
19	Tae	176,48	61,44	323,14	60,81	0,63	-146,66	43,00	43,08	18,36	133,48
20	Vehicle	207,91	68,88	555,77	64,89	3,99	-347,86	2125,97	71,52	-2,64	-1918,06
21	Wdbc	35,85	95,90	183,70	95,14	0,76	-147,85	356,12	96,31	-0,41	-320,27
22	Wine	46,79	98,51	170,94	93,98	4,53	-124,15	80,00	96,60	1,91	-33,21
23	Wisconsin	73,66	96,80	328,02	96,46	0,34	-254,36	521,10	96,35	0,45	-447,44
Trung bình		**114,07**	**82,79**	**355,23**	**80,66**			**281,49**	**80,34**		

Bảng V
SO SÁNH ĐỘ CHÍNH XÁC CỦA HỆ PHÂN LỚP **FRBC_S** SO VỚI **PAES-RCS** VÀ **FURIA** BẰNG PHƯƠNG PHÁP KIỂM ĐỊNH WILCOXON VỚI $\alpha = 0,1$

So sánh (α = 0,1)	R^+	R^-	Exact P-value	Giả thuyết H_0
FRBC_S vs **PAES-RCS**	275,0	1,0	2,622E-5	Bị bác bỏ
FRBC_S vs **FURIA**	227,0	49,0	0,005414	Bị bác bỏ

Bảng VI
SO SÁNH ĐỘ PHỨC TẠP CỦA HỆ PHÂN LỚP **FRBC_S** SO VỚI **PAES-RCS** VÀ **FURIA** BẰNG PHƯƠNG PHÁP KIỂM ĐỊNH WILCOXON VỚI $\alpha = 0,1$

So sánh (α = 0,1)	R^+	R^-	Exact P-value	Giả thuyết H_0
FRBC_S vs **PAES-RCS**	275,0	1,0	4,768E-7	Bị bác bỏ
FRBC_S vs **FURIA**	225,0	51,0	0,00671	Bị bác bỏ

tập kiểm tra cao hơn hệ phân lớp **PAES-RCS** và **FURIA** lần lượt là 21 và 15 trên 23 tập dữ liệu được thử nghiệm. Xét trên giá trị trung bình của độ chính xác phân lớp, hệ phân lớp **FRBC_S** có giá trị trung bình là 82,79%, cao hơn lần lượt là 2,13% và 2,45% so với hệ phân lớp **PAES-RCS** và **FURIA** có giá trị trung bình lần lượt là 80,66% và 80,34%. Phân tích trên độ phức tạp của hệ phân lớp, hệ phân lớp **FRBC_S** có độ phức tạp phân lớp thấp hơn rất nhiều so với hai hệ phân lớp còn lại, tương ứng là 114,07 so với 355,23 và 281,49.

Các kết quả kiểm định giả thuyết thống kê Wilcoxon với độ tin cậy 90% ($\alpha = 0,1$) sử dụng dữ liệu trong Bảng IV đối với độ chính xác phân lớp và độ phức tạp của hệ phân lớp được thể hiện tương ứng trong Bảng V và Bảng VI. Ta thấy rằng, các giá trị giá trị *Exact p-value* đều nhỏ hơn $\alpha = 0,1$ nên giả thuyết tương đương về độ chính xác phân lớp và độ phức tạp của hệ phân lớp của **FRBC_S** tương ứng so với hai hệ phân lớp được đối sánh **PAES-RCS** và **FURIA** bị bác bỏ. Do đó, ta có thể khẳng định rằng hệ phân lớp **FRBC_S** tốt hơn hai hệ phân lớp còn lại trên cả hai tiêu chí độ chính xác phân lớp và độ phức tạp của hệ phân lớp. Do hệ phân lớp **FRBC_S_k0** sử dụng cấu trúc phân hoạch k_0 tốt hơn so với hệ phân lớp **FRBC_S** sử dụng phân hoạch k_1 như đã được so sánh ở trên nên ta có thể kết luận rằng hệ phân lớp **FRBC_S_k0** tốt hơn hai hệ phân lớp **PAES-RCS** và **FURIA**.

IV. KẾT LUẬN

Ngữ nghĩa định tính của các từ ngôn ngữ trong cơ sở luật của hệ phân lớp dựa trên luật mờ không dùng để tính

toán được. Do đó, việc biểu diễn ngữ nghĩa tính toán phù hợp với ngữ nghĩa định tính của các từ ngôn ngữ đóng vai trò quan trọng. Bài báo này trình bày phương pháp biểu diễn ngữ nghĩa tính toán dựa trên tập mờ dạng hàm *S* được sinh ra bởi ĐSGTMR cho các từ ngôn ngữ được sử dụng để biểu diễn cấu trúc phân hoạch đa thể hạt dạng k_0 và k_1. Các kết quả thực nghiệm và kiểm định giả thuyết thống kê Wilcoxon cho thấy tính hiệu quả của các phương pháp được đề xuất khi áp dụng cho hệ phân lớp dựa trên luật mờ.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Trường Đại học Giao thông vận tải trong đề tài mã số T2020-CN-002.

TÀI LIỆU THAM KHẢO

[1] R. Alcalá, Y. Nojima, F. Herrera, H. Ishibuchi, "Multi-objective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions," *Soft Computing*, vol. 15, no. 12, pp. 2303–2318, 2011.

[2] M. Antonelli, P. Ducange, F. Marcelloni, "A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers," *Information Sciences*, vol. 283, pp. 36–54, 2014.

[3] H. Ishibuchi, T. Yamamoto, "Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59-88, 2014.

[4] H. Ishibuchi, T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 428–435, 2005.

[5] M. I. Rey, M. Galende, M. J. Fuente, G. I. Sainz-Palmero, "Multi-objective based Fuzzy Rule Based Systems (FRBSs) for trade-off improvement in accuracy and interpretability: A rule relevance point of view," *Knowledge-Based Systems*, vol. 127, pp. 67–84, 2017.

[6] N. C. Ho, W. Wechler, "Hedge algebras: an algebraic approach to structures of sets of linguistic domains of linguistic truth variables," *Fuzzy Sets and Systems*, vol. 35, no. 3, pp. 281-293, 1990.

[7] N. C. Ho, W. Wechler, "Extended hedge algebras and their application to fuzzy logic," *Fuzzy Sets and Systems*, vol. 52, pp. 259–281, 1992.

[8] N. C. Ho, N. V. Long, "Fuzziness measure on complete hedges algebras and quantifying semantics of terms in linear hedge algebras," *Fuzzy Sets and Systems*, vol. 158, pp. 452-471, 2007.

[9] N. C. Ho, W. Pedrycz, D. T. Long, T. T. Son, "A genetic design of linguistic terms for fuzzy rule based classifiers," *International Journal of Approximate Reasoning*, vol 54, no. 1, pp. 1-21, 2013.

[10] N. C. Ho, T. T. Son, P. D. Phong, "Modeling of a semantics core of linguistic terms based on an extension of hedge algebra semantics and its application," *Knowledge-Based Systems*, vol. 67, pp. 244–262, 2014.

[11] N. C. Ho, H. V. Thong, N. V. Long, "A discussion on interpretability of linguistic rule based systems and its application to solve regression problems," *Knowledge-Based Systems*, vol. 88, pp. 107–133, 2015.

[12] T. T. Son, N. T. Anh, "Partition fuzzy domain with multi-granularity representation of data based on hedge algebra approach," *Journal of Computer Science and Cybernetics*, vol. 34, no. 1, pp. 63–75, 2018.

[13] B. H. Le, L. T. Anh, B. V. Binh, "Explicit formula of hedge-algebras-based fuzzy controller and applications in structural vibration control," *Applied Soft Computing*, vol. 60, pp. 150–166, 2017.

[14] N. H. Huy, N. C. Ho, N. V. Quyen, "Multichannel image contrast enhancement based on linguistic rule-based intensificators," *Applied Soft Computing Journal*, vol. 76, pp. 744–762, 2019.

[15] D. T. Long, "A genetic algorithm based method for timetabling problems using linguistics of hedge algebra in constraints," *Journal of Computer Science and Cybernetics*, vol. 32, no. 4, pp. 285–301, 2016.

[16] J. Demˇsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[17] P. D. Phong, N. C. Ho, N. T. Thuy, "Multi-objective Particle Swarm Optimization Algorithm and its Application to the Fuzzy Rule Based Classifier Design Problem with the Order Based Semantics of Linguistic Terms," *In Proceedings of The 10th IEEE RIVF International Conference on Computing and Communication Technologies (RIVF-2013), Hanoi, Vietnam*, pp. 12–17, 2013.

[18] F. Rudzinski, "A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers," *Applied Soft Computing*, vol. 38, pp. 118–133, 2016.

[19] M. Elkanoa, M. Galara, J. Sanza, H. Bustince, "CHI-BD: A fuzzy rule-based classification system for Big Data classification problems," *Fuzzy Sets and Systems*, vol. 348, pp. 75–101, 2018.

[20] M. Pota, M. Esposito, G. D. Pietro, "Designing rule-based fuzzy systems for classification in medicine," *Knowledge-Based Systems*, vol. 124, pp. 105–132, 2017.

[21] M. Soui, I. Gasmi, S. Smiti, K. Ghédira, "Rule-based credit risk assessment model using multi-objective evolutionary algorithms," *Expert Systems With Applications*, vol. 126, pp. 144–157, 2019.

[22] Y. Zhang, X. Qian, J. Wang, M. Gendeel1, "Fuzzy rule-based classification system using multi-population quantum evolutionary algorithm with contradictory rule reconstruction," *Applied Intelligence*, vol. 49, pp. 4007–4021, 2019.

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Đức Dư

Nhận bằng Cử nhân Toán tin ứng dụng, Thạc sĩ Toán ứng dụng tại Trường Đại học khoa học tự nhiên, Đại học Quốc gia Hà Nội lần lượt các năm 2001, 2005. Hiện là giảng viên Khoa Công nghệ thông tin, Trường Đại học Giao thông vận tải. Lĩnh vực nghiên cứu: khai phá dữ liệu, lô gic mờ, hệ mờ, tính toán mềm, tính toán với từ, học máy.

Phạm Đình Phong

Nhận bằng Thạc sĩ Công nghệ thông tin và Tiến sĩ Khoa học máy tính tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội lần lượt các năm 2011, 2018. Hiện là giảng viên Khoa Công nghệ thông tin, Trường Đại học Giao thông vận tải. Lĩnh Vực nghiên cứu: khai phá dữ liệu, các hệ mờ, tính toán mềm, học máy.

Phạm Đình Vũ

Nhận bằng kỹ sư Công nghệ thông tin tại Trường Đại học Bách khoa Hà Nội năm 2003, Thạc sỹ Hệ thống thông tin Học viện Công nghệ Bưu chính Viễn thông năm 2015. Hiện đang công tác tại Cục Công nghệ thông tin và Thống kê hải quan, Tổng cục Hải quan. Lĩnh vực nghiên cứu: khai phá dữ liệu, các hệ mờ, tính toán mềm, học máy.

Nguyễn Đức Thảo

Nhận bằng Kỹ sư, Thạc sĩ và Tiến sỹ Công nghệ thông tin lần lượt tại Trường Đại học Tổng hợp Nga năm 1996, 2001. Hiện là cán bộ nghiên cứu tại Viện Khoa học và Công nghệ quân sự/ Bộ Quốc phòng. Lĩnh vực nghiên cứu: khai phá dữ liệu, lô gic mờ, hệ mờ, tính toán mềm, tính toán với từ, học máy, trí tuệ nhân tạo, hệ thống thông tin, hệ chuyên gia.

Bus-RSN: Giải pháp tô-pô mạng liên kết dạng lai cho các trung tâm dữ liệu cỡ vừa, tiết kiệm chi phí và đáp ứng không gian mở

Kiều Thành Chung[1, 3], Vũ Quang Sơn[2], Phạm Đăng Hải[3], Nguyễn Khanh Văn[3]
[1]Trường Cao đẳng nghề Công nghệ cao Hà Nội
[2]Học viện Công nghệ Bưu chính Viễn thông
[3]Trường Đại học Bách khoa Hà Nội

Tác giả liên hệ: Kiều Thành Chung, kieuthanhchung@gmail.com
Ngày nhận bài: 08/05/2020, ngày sửa chữa: 21/06/2020
Định danh DOI: 10.32913/mic-ict-research-vn.v2020.n1.922

Tóm tắt: Xây dựng và phát triển các trung tâm dữ liệu (DC – Data Center) kích thước vừa và nhỏ đang được các doanh nghiệp Việt nam quan tâm. Một thách thức lớn ở đây là thiết kế tô-pô liên kết có chi phí triển khai thấp mà đảm bảo tính linh hoạt cao như dễ mở rộng và đáp ứng điều kiện hạn chế về không gian như: kết hợp nhiều phòng máy chủ có diện tích sàn bị giới hạn, không liền kề (đôi khi ở các tầng khác nhau trong tòa nhà). Các kiến trúc tô-pô mạng liên kết được ứng dụng phổ biến trên thế giới là không thực sự phù hợp cho những điều kiện thực tế đặc thù này. Trong bài báo này, chúng tôi đề xuất một giải pháp tô-pô mạng liên kết dạng lai, ký hiệu Bus-RSN, có thiết kế đặc thù phù hợp cho việc lắp đặt các DC cỡ vừa và nhỏ nhằm thỏa mãn các mục tiêu trên. Tô-pô đề xuất được tạo ra bằng cách lai kết hợp tô-pô dạng Bus và tô-pô RSN (Random Shortcut Networks) [8], trong đó thành phần bus đóng vai trò là đường trục kết nối các vùng máy chủ liên tục (tương ứng một phòng hay sàn) mà mỗi vùng được liên kết bằng một cấu trúc RSN, phù hợp với giới hạn không gian riêng biệt của mỗi phòng máy chủ. So sánh với các giải pháp tô-pô hiện có đáng quan tâm thông qua thực nghiệm với công cụ phần mềm [1], chúng tôi thấy giải pháp đề xuất có thể đem lại một lựa chọn có tính thực tiễn cao: giảm được chi phí thiết bị đáng kể trong khi yếu tố hiệu năng quan trọng nhất (đỗ trễ truyền tin) bị ảnh hưởng giảm tương đối nhỏ. Chẳng hạn với một cấu trúc không gian gồm 2 sàn lắp đặt riêng biệt, Bus-RSN có thể tiết kiệm chi phí thiết bị mạng đến 26% so với tô-pô hiện đại hàng đầu là JellyFish [2] mà chỉ thua kém 12% về độ trễ truyền tin.

Từ khóa: *tô-pô mạng, mạng liên kết dạng lai, thuật toán định tuyến, mạng ngẫu nhiên.*

Title: **Bus-RSN: An Interconnect Topology for Medium-Size Data Centers, Saving in Cable and Fitting to Incremental Expansion to Non-continuous Space**

Abstract: The demand of building Industrial Data Centers in small and medium sizes is growing significantly in Vietnam. One faces a challenging task in designing interconnection topologies that are flexible and incremental, of initial cheap cost, and suitable for being deployed in separate rooms. Existing popular interconnect topologies are not good enough in these mentioned special conditions. In this paper, we propose a hybrid interconnect, dubbed Bus-RSN, that is specially designed for these mentioned purposes. Simplistically speaking, this interconnect is created by combining a Bus structure and a random shortcut network (RSN [8]) wherein the bus is used as a backbone to connect the separate server rooms while each server room is locally connected by a RSN structure. Using our simulation-based software tool [1] we compare our proposed interconnect with popular suitable existing ones, including JellyFish [2], and R3 [3], and obtain encouraging results: ours loses a bit in latency (12%) but saves a lot in cable cost (26%) comparing to JellyFish, the popular for fast communication.

Keywords: *network topology, hybrid-network, routing algorithm, random network.*

I. GIỚI THIỆU

Nhu cầu xây dựng và phát triển riêng các hệ thống DC (Data Center, viết tắt: DC) cỡ vừa và nhỏ tại một số doanh nghiệp Việt nam đang trở nên phổ biến, ví dụ như tại các ngân hàng, thư viện dữ liệu số, trung tâm báo chí. Ngoài phương án thuê địa điểm đặt máy chủ tại các DC lớn của các doanh nghiệp công nghệ hàng đầu như Viettel, VNPT,

FPT . . . , các doanh nghiệp đủ lớn cũng đang mở rộng thêm phương án đầu tư xây dựng mới các DC cỡ nhỏ (khoảng vài chục đến vài trăm máy chủ), nhằm chủ động trong việc đầu tư, khai thác hệ thống. Ban đầu các DC này có thể chỉ phục vụ hoạt động riêng của doanh nghiệp nhưng sau đó nó có thể mở rộng lên khi doanh nghiệp mở rộng kinh doanh và có thể kết hợp cho bên ngoài thuê bao khai thác. Vì thế, nét đặc thù của xu hướng này là việc thiết lập không gian cho DC cần mang tính linh hoạt, mềm dẻo để có thể mở rộng dần, thích hợp với việc các DC có thể tăng trưởng kích thước liên tục (từ cỡ nhỏ, vài chục đến hàng trăm máy, lên cỡ vừa, hàng nghìn máy).

Khác với việc xây dựng các trung tâm dữ liệu lớn thường bao gồm việc xây dựng những tòa nhà hay khu vực không gian biệt lập với các điều kiện lý tưởng, việc lập phương án xây dựng các DC vừa và nhỏ thường ưu tiên tính kinh tế bằng cách khai thác các khoảng không gian còn trống hay được chuyển đổi chức năng trong các tòa nhà doanh nghiệp đã có sẵn. Những không gian mang tính huy động gom lại này thường có những hạn chế như không đủ rộng lớn và liên tục, có thể là sự kết hợp của nhiều phòng hay mặt sàn sử dụng tách rời nhau (thậm chí có thể nằm trên nhiều tầng tách biệt của một tòa nhà). Ở một số tình huống khác, một DC cỡ nhỏ ban đầu có thể được phát triển dần dần về qui mô và có thể "phình ra" vượt quá sự cho phép của không gian thiết kế ban đầu. Thường là do khả năng huy động nguồn vốn ban đầu, DC có thể có kích thước nhỏ và bố trí lắp đặt trong một không gian vừa phải, nhưng nếu sau đó hoạt động hiệu quả, doanh nghiệp có thể huy động thêm vốn và muốn tìm cách mở rộng kích thước DC một cách linh hoạt. Do đó không gian lắp đặt ban đầu có thể không còn chỗ và phải tiến hành xem xét việc mở rộng sang khu vực mới (mặt sàn mới) và cần tìm phương án giải quyết việc kết nối giữa 2 khu vực sao cho hiệu quả nhất.

Các DC có thể được mô hình hóa dưới dạng đồ thị $G(V, E)$ với G là tập các đỉnh biểu diễn cho các nút mạng và E là tập các cạnh biểu diễn cho các liên kết giữa chúng, được tạo ra bởi các luật liên kết được định trước. Trong bài báo này, với $N = |V|$ đỉnh cho trước, chúng tôi nghiên cứu tìm cách sắp xếp N đỉnh này vào không gian không liên tục và luật liên kết giữa chúng để xây dựng một mô hình mạng liên kết phù hợp nhất, tức là đạt được hiệu năng và tính kinh tế thích hợp, cho dạng DC có tính *dễ mở rộng tăng dần* đồng thời có thể *bố trí lắp đặt trong một không gian thiếu liên tục,* tức là có thể là sự kết hợp của nhiều mặt sàn (không gian lắp đặt liên tục) tách rời và có khoảng cách nhất định giữa chúng. Chúng tôi trước hết khảo sát các mô hình mạng liên kết đã được đề xuất thuộc thiết kế tô-pô mạng liên kết hiệu năng cao, từ đó tìm cách phát triển một dạng tô-pô kiểu mới phù hợp với thực trạng doanh nghiệp nhỏ và vừa.

Theo kết quả của một quá trình khảo sát tương đối công phu chúng tôi nhận thấy có 3 cách tiếp cận chính trong thiết kế tô-pô mạng liên kết của một DC: theo dạng tô-pô chuẩn tắc (regular topology), hoặc tô-pô mạng ngẫu nhiên (random network topology), hoặc tô-pô lai (hybrid topology) là sự kết hợp những đặc thù từ cả hai dạng trước đó.

Các tô-pô dạng chuẩn tắc phải tuân theo các qui ước nghiêm ngặt như tính đối xứng giữa các đỉnh, hay các tính chất cấu trúc đặc biệt riêng nên việc mở rộng qui mô là gần như không thể; nói cách khác, các mạng liên kết đã được thiết kế theo tô-pô dạng chuẩn tắc là sẽ gần như đóng với mọi sự thay đổi. Với các tô-pô chuẩn tắc truyền thống, nếu như bắt buộc phải thay đổi để tăng qui mô, gần như người ta sẽ phải xây dựng lại cả hệ thống mạng. Các tô-pô thế hệ mới, được đề xuất gần đây như Fat-Tree [4, 5], có một số cải thiện về khả năng mở rộng tuy nhiên vẫn còn kém linh hoạt do vẫn cần đảm bảo tính đối xứng và bậc của nút mạng là khá lớn. Muốn mở rộng qui mô, doanh nghiệp sẽ cần phải thay thế toàn bộ các thiết bị chuyển mạch cũ bằng loại mới với số cổng (port) cao hơn, thường là đắt tiền hơn nhiều, dẫn đến chi phí mở rộng rất lớn. Ngoài ra do Fat-Tree có bậc đỉnh cao, yêu cầu xây dựng DC trong không gian là kết hợp của một số mặt sàn rời nhau sẽ gây rất tốn kém về cáp mạng.

Tiếp cận tô-pô dạng mạng ngẫu nhiên ra đời chính nhằm mục đích đảm bảo khả năng mở rộng qui mô tăng dần (incremental growth), trong đó tô-pô JellyFish [2] là một điển hình và được giới học thuật quan tâm nhiều. Tuy nhiên các tô-pô dạng này vẫn ít nhiều có tính đối xứng và bậc đỉnh cao (JellyFish là một đồ thị ngẫu nhiên có bậc đỉnh hằng số r cho trước) nên cũng đòi hỏi huy động cáp mạng rất lớn nếu không gian lắp đặt không là một mặt sàn liên tục duy nhất.

Một nhóm các giải pháp tô-pô lai cũng đã được đề xuất, với tô-pô dựa trên cấu trúc siêu khối (hyper-cube) và thực hiện đệ quy để mở rộng, ví dụ R3 [3], BCN [6], DCell [7]. Do dựa trên cấu trúc siêu khối, nên một hạn chế đáng kể của các dạng tô-pô này chính là sự mở rộng, ví dụ, nếu bổ sung thêm một vài nút mạng trong một tô-pô thành phần, thì các tô-pô thành phần khác cũng phải được bổ sung thêm số lượng nút tương ứng để đảm bảo tính chuẩn và tính đệ quy. Điều này có thể gây ra sự dư thừa nút mạng, gia tăng không cần thiết về chi phí thiết bị và triển khai cài đặt mở rộng. Hơn nữa, các tô-pô thành phần phải được triển khai trên các phòng máy chủ có không gian đồng đều nhau.

Theo khi khảo sát kỹ các tô-pô mạng liên kết đã biết theo cả 3 nhóm tiếp cận nói trên, đặc biệt chú ý khảo sát các đại biểu của mỗi nhóm là Fat-Tree [4], JellyFish [2] và R3 [3], chúng tôi rút ra kết luận là các thiết kế tô-pô mạng được đề xuất trước đây hướng tới việc xây dựng các DC

cỡ lớn và mặc dù không đề cập đến vấn đề không gian lắp đặt của các phòng máy chủ, có một giả thiết ngầm định là các cơ sở vật chất (không gian lắp đặt) được xây dựng sao cho phù hợp với thiết kế tô-pô mạng tương ứng. Do đó hầu hết các tô-pô mạng phổ biến, bao gồm cả các tô-pô lai loại hiện đại hướng tới tính mở rộng, sẽ gặp khó khăn lớn trong việc triển khai trên một địa bàn có tính chất thiếu liên tục và không đồng nhất như đề cập trên.

Trong bài báo này, chúng tôi đề xuất Bus-RSN, một mô hình tô-pô lai (hybrid topology) như là một giải pháp chuyên dụng, đề xuất riêng cho bài toán xây dựng DC với những yếu tố đặc thù nói trên. Kiến trúc này là sự kết hợp hai dạng tô-pô: 1) Bus – cổ điển và tối giản, và 2) RSN (Random Shortcut Networks) [8]; sự ra đời của nó là xuất phát từ những quan sát thực tế phát triển DC trong nước của chúng tôi. Với một trung tâm DC cỡ vừa hoặc nhỏ, có thể lắp đặt trong không gian hợp thành từ nhiều phòng (sàn) rời rạc; các ứng dụng khai thác cũng thường yêu cầu một số lượng máy chủ phục vụ không lớn, phù hợp triển khai trong một phòng máy chủ nào đó. Vì vậy giao thông[1] qua lại của một ứng dụng khai thác thường xuyên sẽ là giữa các máy chủ cùng phòng, là không lớn. Do đó, chúng tôi đề xuất sử dụng cấu trúc Bus như là một "đường xương sống" để liên kết các phòng máy chủ, đủ cho các nhiệm vụ quản lý, kiểm soát mạng đồng thời thiết kiệm chi phí cáp mạng và thiết bị liên quan. Đối với mỗi phòng máy chủ, chúng tôi áp dụng mô hình liên kết ngẫu nhiên RSN (phối hợp giữa cấu trúc lưới và các liên kết ngẫu nhiên) để làm giảm đường kính mạng và vẫn cho phép tiết kiệm cáp mạng. Bên cạnh đó, chúng tôi cũng thêm vào thiết kế một dạng liên kết ngẫu nhiên giữa các phòng máy, được bổ sung vào các nút mạng đặc biệt (gọi là bus-node) giữ nhiệm vụ như các nút trung tâm của các khối đơn vị thành phần của các mặt sàn phòng máy liên tục (chúng tôi dùng các thuật ngữ block và zone để gọi các cấu thành mạng nói trên). Những liên kết ngẫu nhiên loại này cho phép rút ngắn đường kính mạng (graph diameter) và đảm bảo rằng ngay cả khi các ứng dụng triển khai trên nhiều hơn một mặt sàn (zone) thì vẫn được phục vụ hiệu quả.

Thông qua đánh giá bằng thực nghiệm, chúng tôi thấy BUS-RSN đạt được những hiệu quả tốt hơn đáng khích lệ khi so sánh với những tô-pô cụ thể đại biểu quan trọng của các nhóm nói trên (Fat-Tree [4, 5], JellyFish [2], R3 [3]) khi cùng thiết lập với những điều kiện đặc thù cụ thể đã đề cập. Chẳng hạn như đối với một yêu cầu thiết lập mạng 128 nút chuyển mạch (switch), lắp đặt trên 2 mặt sàn (zone) biệt lập (cách nhau 15 mét) mỗi sàn có 4 khối đơn vị (block), Bảng I cho thấy, thiết lập theo JellyFish (JF-KSPR-2) sẽ đạt được hiệu năng cao nhất về độ trễ ước lượng (Estimated Latency) nhưng hơn thiết lập theo BUS-RSN (BR-HRA-2-4) đứng thứ hai chỉ khoảng 12%; trong khi đó BR-HRA-2-4 là tiết kiệm nhất về cáp mạng, chỉ là 18% khi so với JF-KSPR-2, và do đó tổng giá thành dự kiến chỉ là 74% của JF-KSPR-2.

Bảng I
KẾT QUẢ THỰC NGHIỆM BUS-RSN ĐƯỢC THIẾT LẬP 2 ZONE, MỖI ZONE 4 BLOCKS TRÊN MẠNG 128 NÚT.

128 nút mạng	ARPL	Latency	Cable	Total cost
BR-HRA-2-4	2,95	326,45	1.528,20	731.804,30
JF-KSPR-2	2,90	289,89	8.273,20	986.491,50
R3-SPR-8	4,78	534,62	3.605,2	826.216,60
RSN-SPR-2	3,12	344,28	3.813,00	784.523,55

Rõ ràng những kết quả thực nghiệm như thế này cho thấy giải pháp đề xuất của chúng tôi là rất đáng được xem xét và có thể là một lựa chọn hữu ích cho các doanh nghiệp trong nước đang xem xét thiết kế DC với các điều kiện đặc thù đề cập.

Phần còn lại của bài báo được tổ chức như sau: phần II trình bày các nghiên cứu liên quan, phần III trình bày giải pháp chính, phần IV trình bày các kết quả thực nghiệm cùng với một số thiết lập mạng cụ thể theo điều kiện đặc thù quan tâm và phần kết luận được trình bày trong phần V.

II. CÁC NGHIÊN CỨU LIÊN QUAN

1. Tô-pô mạng liên kết

Cấu trúc tô-pô thể hiện sự sắp đặt các nút mạng[2] và các liên kết giữa chúng. Trong phạm vi bài báo này, chúng tôi tập trung vào tô-pô áp dụng cho DC mà trong đó các máy chủ được kết nối với nhau thông qua cấu trúc mạng chuyên dụng, sao cho đảm bảo các mục tiêu thiết kế cụ thể như là chi phí thiết bị thấp, khả năng đáp ứng cao, và mở rộng nhanh chóng.

Các tô-pô truyền thống trước đây được đề xuất áp dụng cho DC với kích thước mạng nhỏ, ví dụ Grid, Torus, HyberCube. Các dạng tô-pô này được gọi là các tô-pô chuẩn tắc với các quy tắc chặt chẽ về mặt kết nối và đảm bảo bậc đỉnh đều trên mọi nút. Tuy nhiên, khi kích thước mạng (số nút mạng) tăng lên nhanh chóng do sự phát triển mạnh của DC, các tô-pô truyền thống trở nên kém hiệu quả khi không đáp ứng được khả năng mở rộng và tính mềm dẻo. Cụ thể là việc bổ sung thêm các nút mạng, tăng/giảm kích thước mạng có thể được tiến hành thường xuyên, nhiều lần với quy mô đa dạng, nhưng vẫn phải đảm bảo hiệu năng mạng. Với kích thước mạng ngày càng tăng, tính mềm dẻo đòi hỏi cao thì các vấn đề như tiết kiệm chi phí thiết bị và cáp kết nối, tiết kiệm năng lượng khi tải thấp, trở thành những chủ

[1] Sự di chuyển của các gói tin giữa các nút mạng

[2] ví dụ như các máy tính toán hoặc các thiết bị chuyển mạch – switch

đề đáng quan tâm. Có thể thấy vấn đề này được đề cập trong hàng loạt nghiên cứu gần đây về việc thiết kế mạng liên kết trong DC như HELIOS [9], BCube [10], Dcell [7], Scafida [11], MDCube [12], VL2 [13]. Do vậy, các nhà nghiên cứu tích cực đề xuất những dạng kiến trúc tô-pô mới phù hợp hơn cho các yêu cầu hiện đại như Smallworld DataCenter [14], JellyFish [2], Space Shuffle [15], Fat-Tree [4, 5], SkyWalk [16], Dragonfly [17].

Một vài đề xuất gần đây đối với các mạng dung lượng cao khai thác cấu trúc cụ thể cho tô-pô và thuật toán định tuyến tương ứng. Chúng bao gồm Folded-Clos (hay còn gọi là Fat-Tree) [4, 5, 13, 18], một vài thiết kế có sử dụng các máy chủ trong việc chuyển tiếp gói tin [7, 10, 12] và các thiết kế sử dụng công nghệ mạng kết nối quang [9, 19]. Đối với một số tô-pô cụ thể, việc bổ sung thêm số lượng máy chủ trong khi phải đảm bảo các thuộc tính chặt chẽ của cấu trúc sẽ phải thay thế một số lượng lớn các thành phần mạng cũng như thực hiện lại việc thi công lại tuyến cáp. MDCube [12] cho phép mở rộng mạng thêm vài nghìn máy chủ, trong khi DCell [7] và BCube [10] cho phép mở rộng đến các kích thước mạng dự kiến, nhưng phải thiết kế bổ sung các cổng kết nối dự phòng, điều này dẫn đến sự dư thừa và chi phí lắp đặt tăng lên.

Trong số các đề xuất, Fat-Tree [4, 5] được xem là một trong những mô hình hiệu quả đối với DC có kích thước mạng lớn (ví dụ, có thể hỗ trợ kết nối lên đến 65536 máy chủ với thiết bị switch có 64 cổng) với việc tổ chức cấu trúc mạng thành 3 tầng riêng biệt, bao gồm tầng lõi (Core Layer), tầng trung gian (Aggregation Layer) và tầng kết nối trực tiếp các host (Edge Layer). Mặc dù bậc đỉnh của các nút mạng trong mô hình Fat-Tree không đều nhau, tuy nhiên, xét trong cùng một tầng, các nút có cùng bậc đỉnh và chúng được kết nối với nhau theo quy luật một cách chặt chẽ. Do vậy, Fat-Tree được xem là một regular topology thế hệ sau so với các Regular Topology truyền thống (ví dụ như Torus, Grid, HyperCube). Thiết kế của Fat-Tree đảm bảo tham số oversubcription đạt 1:1, có nghĩa là đảm bảo tất cả các host có khả năng kết nối tới bất kỳ host khác với băng thông đường truyền kết nối đạt tối đa. Mặc dù Fat-Tree có nhiều ưu điểm như đường kính mạng thấp, độ trễ truyền tin thấp, *oversubcription*[3] đạt 1:1, tuy nhiên, nó cũng tồn tại một số hạn chế đáng kể. Do cấu trúc chặt chẽ, nên khả năng mở rộng của Fat-Tree phụ thuộc vào số cổng thiết bị switch được sử dụng trong việc kết nối mạng. Ví dụ, kích thước 3456, 8192, 27648 và 65536 tương ứng với số cổng switch là 24, 32, 48 và 64. Trong trường hợp giả định có thể sử dụng switch 50 cổng, có thể bổ sung thêm 3602 máy chủ nhưng phải thay thế toàn bộ các thiết bị switch ban đầu (48 cổng). Đây chính là hạn chế về khả năng mở rộng và làm gia tăng chi phí thiết bị của Fat-Tree.

Một cách tiếp cận mới và hấp dẫn gần đây là sử dụng các mô hình mạng ngẫu nhiên trong việc thiết kế các tô mô mạng liên kết mà có thể đạt hiệu quả để cung cấp chất lượng hiệu năng quan trọng như là tính linh hoạt (flexibility), tính mở rộng (scalability) và tính thích nghi (adaptivity) mà chúng là các đòi hỏi quan trọng đối với các tô-pô DC hiện đại như đã được đề cập. Thông thường, tô-pô ngẫu nhiên được xây dựng bằng việc bổ sung thêm các liên kết ngẫu nhiên đối với một đồ thị cơ bản đơn giản và chính tắc như dạng lưới 2-D; các liên kết ngẫu nhiên được tạo ra bởi một phân bố xác suất nào đó, nhưng thường là phân bố đều. Ví dụ như đồ thị liên kết ngẫu nhiên RSN, được tạo ra bởi việc bổ sung các liên kết ngẫu nhiên giữa các nút trên đồ thị cơ sở dạng lưới (grid). Mô hình xây dựng mạng này được chỉ ra tại [8], có thể đạt được đường kính đồ thị (và độ trễ truyền thông) giảm đáng kể so với các tô-pô mạng truyền thống (với cùng kích thước và cùng bậc đỉnh). Hơn nữa, các tính chất quan trọng của việc mở rộng một cách tự nhiên với mô hình mạng ngẫu nhiên, do đó, nó hấp dẫn đối với mạng DC Jellyfish [2], Scafida [11]. Đề xuất JellyFish sử dụng mô hình mạng ngẫu nhiên để hỗ trợ mở rộng mạng và giá trị oversubcription đạt tỉ lệ 1:1. Tuy nhiên, JellyFish phải sử dụng thuật toán định tuyến -SPR với kích thước bảng định tuyến lớn và sử dụng các switch có bậc đỉnh cao (ví dụ, 48 cổng).

Mặc dù cả Fat-Tree [4, 5] và JellyFish [2] đều đạt được giá trị oversubcription 1:1, nhưng bậc đỉnh của hai đại diện này khá cao, dẫn đến chi phí thiết bị mạng cao. Fat-Tree còn tồn tại bất lợi trong việc mở rộng mạng trong khi vẫn phải đảm bảo các tính chất chặt chẽ của cấu trúc mặc dù có lợi thế trong việc thực hiện định tuyến dễ dàng. Ngược lại, JellyFish có nhược điểm về việc xây dựng luật định tuyến do tính chất ngẫu nhiên của các liên kết, tuy nhiên, JellyFish lại có khả năng hỗ trợ mở rộng mạng (đảm bảo tính co giãn quy mô). Vậy có tồn tại một mô hình tô-pô có thể kết hợp và khai thác được những ưu điểm của những tô-pô trên (regular và ir-regular), đồng thời hạn chế những nhược điểm của chúng?

Một dạng mô hình với ý tưởng dựa trên tô-pô lai đang được quan tâm hiện nay là đồ thị hỗn hợp (compound graph). Compound graph cấp 1 $G(G1)$ là mạng liên kết được tạo ra bởi 2 regular graph G và $G1$, bằng cách thay thế các nút của G bằng $G1$ và các liên kết của G được thay thế bởi các liên kết mới giữa 2 bản sao của $G1$ tương ứng với 2 nút của G mà nó thay thế. Compound graph cấp cao hơn có thể được tạo ra từ các compound cấp thấp theo

[3]Các thiết kế DC đề xuất oversubcription như là một tham số để làm giảm tổng chi phí của thiết kế đó. Oversubcription là tỉ lệ giữa tổng băng thông, trong trường hợp tồi nhất, có thể đạt được giữa các hosts đối với tổng băng thông cực đại của một cấu trúc tô-pô cụ thể. Ví dụ, giá trị oversubcription 5:1 có nghĩa là băng thông của host chỉ đạt 20%. Thông thường, các DC được thiết kế oversubcription từ 2,5:1 (tương đương 400Mbps) đến 8:1 (tương đương 125Mbps) đối với băng thông đường truyền là cho mạng sử dụng switch thông thường Ethernet

phương pháp đệ quy. Ngoài việc giữ được tính chất ban đầu của G, compound graph được bổ sung các thuộc tính tốt của $G1$. Vì vậy có nhiều DC được đề xuất theo hướng sử dụng mô hình này, ví dụ BCN [6], DCell [7], R3 [3]... Trong đó, R3 là tô-pô đáng chú ý khi có cấu trúc khá phù hợp với bài toán mà chúng tôi đưa ra.

R3 là tô-pô được tạo ra dựa trên mô hình compound graph, bằng cách kết hợp 2 tô-pô: generalized hybercube (GHC) và random r-regular graph (RRG, trong R3 được gọi là cluster). Với GHC, R3 khá linh hoạt trong việc xây dựng quy mô DC với số nút mạng cho trước. Ngoài ra, RRG là tô-pô có đặc tính đường kính mạng nhỏ, linh hoạt trong việc mở rộng... Trong khi sở hữu các tính chất trên, R3 vẫn tồn tại một số nhược điểm. Do được tạo bởi GHC nên R3 cần có sự đánh đổi giữa đường kính mạng và bậc của nút. Hơn nữa, do cấu trúc random của RRG và phải liên kết nhiều RRG theo cấu trúc của GHC, nên việc truy vết các liên kết của tô-pô này khá khó khăn. Nếu áp dụng R3 [3] để giải quyết bài toán đã nêu trong phần I thì có thể dẫn đến chi phí dùng cho switch và cáp mạng tăng cao cùng với khó khăn trong quá trình cài đặt và bảo trì.

2. Thuật toán định tuyến

Một vấn đề quan trọng trong việc đề xuất thiết kế tô-pô là xây dựng thuật toán định tuyến tương ứng để khai thác tối đa đặc tính tô-pô nhằm đảm bảo các yếu tố hiệu năng mạng. Quá trình định tuyến xác định đường truyền tin giữa các cặp nguồn – đích trong mạng. Các thuật toán định tuyến SPR sử dụng thuật toán Dijkstra hoặc thuật toán tham lam để xác định đường đi ngắn nhất giữa các cặp nguồn đích. Do tính chặt chẽ của cấu trúc, các tô-pô dạng chuẩn tắc thường dễ dàng áp dụng các thuật toán SPR, ví dụ, 2-D Torus, 3-D Torus, Fat-Tree [4]. Tuy nhiên, đối với các tô-pô ngẫu nhiên, ví dụ RSN, JellyFish, các thuật toán SPR tỏ ra kém hiệu quả với đường kính lớn. Để khắc phục nhược điểm đó, các thuật toán rút gọn (compact routing) được đề xuất áp dụng cho các đồ thị ngẫu nhiên, ví dụ, thuật toán TZ của tác giả Thorup & Zwick [20], CORRA [21] với việc khai thác thông tin định tuyến được lưu trữ tại các nút mạng (bảng định tuyến tại các nút mạng). Ví dụ, thuật toán TZ [20] khai thác thông tin của nút đại diện (landmark) của tập các nút trong vùng mà nó đại diện, bằng cách định tuyến từ nút nguồn tới nút đại diện gần nhất của nút đích, và sau đó chuyển tiếp gói tin tới nút đích; hoặc thuật toán CORRA [21], sử dụng các liên kết dài như là các cầu nối giữa các vùng nút để chuyển tiếp thông tin. Các thuật toán Compact không tuân theo định tuyến ngắn nhất, mà nó khai thác các liên kết giữa các nút xa nhau, từ đó làm giảm đường kính mạng và giảm trung bình độ trễ truyền tin.

Một vấn đề khá khó khăn trong cấu trúc tô-pô lai chính là làm thế nào định tuyến gói tin giữa các nút mạng trong nó. Với mỗi dạng tô-pô thành phần sẽ có kịch bản định tuyến tương ứng nhằm đảm bảo các tham số hiệu năng mạng. Tuy nhiên, việc đưa ra được luật định tuyến chung cho cấu trúc tô-pô lai trở nên khó khi mà các tô-pô regular thường áp dụng các thuật toán định tuyến theo chiều DOR hoặc sử dụng giao thức Duato [22], trong khi đó, các tô-pô ngẫu nhiên tỏ ra hiệu quả đối với kịch bản định tuyến rút gọn (CANDAR [23], CORRA [21], TZ [20]. Các thuật toán SPR tỏ ra kém hiệu quả đối với các tô-pô lai khi phải sử dụng quá nhiều tài nguyên (bộ nhớ tại mỗi switch). Trong trường hợp áp dụng kịch bản định tuyến rút gọn trên toàn bộ cấu trúc tô-pô lai, việc định tuyến trở nên khó khăn khi không xác định được luật định tuyến chung. Trong cả hai trường hợp, việc định tuyến không mang lại hiệu quả tốt. Một chiến lược định tuyến R3 [3], được đề xuất nhằm áp dụng hiệu quả trên dạng tô-pô lai. Ý tưởng chính của cách tiếp cận này là kết hợp các thuật toán định tuyến khác nhau để tối ưu hiệu năng mạng và công việc định tuyến cũng được chia ra thành các trường hợp riêng biệt. Theo đó, R3 được định tuyến với 2 trường hợp: định tuyến nội vùng (intra-cluster, định tuyến giữa các nút mạng trong cùng một vùng) và định tuyến liên vùng (inter-cluster, định tuyến giữa các nút mạng thuộc hai vùng khác nhau). Trong trường hợp intra-cluster, 2 nút thuộc cùng 1 cluster, gói tin được định tuyến bằng thuật toán $K*$ ($K*$ algorithm) [24] (là thuật toán sử dụng hàm mục tiêu heuristic hiệu quả nhất hiện nay, được áp dụng để tìm kiếm k đường đi ngắn nhất). Đối với inter-cluster, 2 nút thuộc 2 cluster khác nhau, thuật toán định tuyến được kết hợp bởi các kỹ thuật khác nhau: Thuật toán Dijkstra và thuật toán định tuyến dựa trên tọa độ của các nút.

Chúng tôi xem xét kỹ hơn và áp dụng cách tiếp cận kết hợp nhiều thuật toán khác nhau khi định tuyến trong mô hình Bus-RSN. Trong đó, chúng tôi áp dụng luật định tuyến 2 giai đoạn tương tự như thuật toán định tuyến rút gọn [20] cho Bus-RSN. Giai đoạn 1, gói tin được định tuyến đến bus-node gần nút đích nhất và giai đoạn 2 là gói tin được đưa đến đích, chi tiết thuật toán được mô tả tại phần III.2

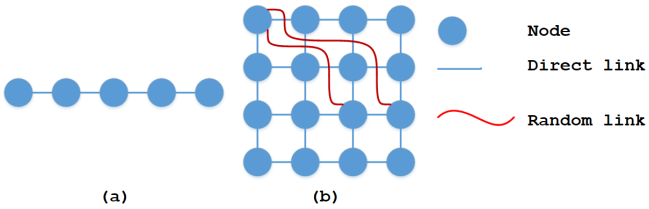
III. GIẢI PHÁP TÔ-PÔ LAI ĐỀ XUẤT

Xét bài toán xây dựng DC với quy mô ban đầu nhỏ, được lắp đặt tại nhiều phòng/sàn riêng biệt, sao cho nó có khả năng linh hoạt trong việc mở rộng mà đảm bảo thông số hiệu năng cao hợp lý.

Ý tưởng cơ bản của chúng tôi là kết hợp lai giữa tô-pô dạng Bus[4] và RSN sao cho khai thác ưu điểm của chúng một cách phù hợp. Trước hết chúng tôi dự định kết nối các phòng/sàn bằng tô-pô dạng Bus do có thể đáp ứng được việc định tuyến và lắp đặt dễ dàng và hơn nữa lại linh hoạt

[4]Mạng liên kết với các nút được kết nối với nhau như 1 đường thẳng, liên kết với các nút liền kề với nó, ngoại trừ nút đầu và nút cuối cùng.

trong việc mở rộng DC trong lương lai: dễ dàng triển khai thêm các phòng máy chủ bằng việc thêm các nút mạng đại diện của các phòng đó vào 2 đầu của đường Bus, như Hình 1-a. Tất nhiên, tô-pô này không cho hiệu năng mạng cao, do đường kính mạng quá lớn, bên cạnh đó, mỗi phòng lại có kích thước khác nhau, có số lượng nút mạng khác nhau; do đó chúng tôi lựa chọn kiến trúc RSN (minh họa như Hình 1-b) cho tô-pô liên kết nội bộ mỗi phòng để vừa đảm bảo tính linh hoạt vừa có đường kính mạng đủ nhỏ.



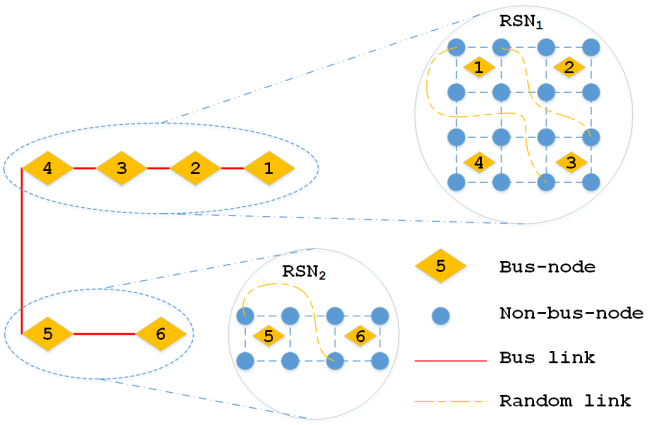
Hình 1. a. Bus topology với 5 node
b. RSN 4x4 tạo bởi grid graph và random link

1. Kiến trúc

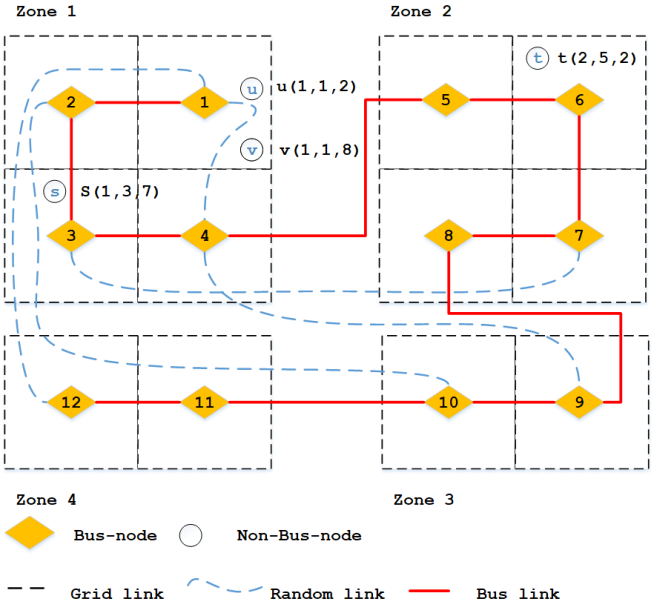
Phần coi là "bus" trong tô-pô Bus-RSN tạo thành một trục dọc "bus-way" liên kết nhiều sàn (không gian lắp đặt của một phòng máy chủ) phân biệt có kích thước khác nhau. Các sàn được ký hiệu là RSN_i (ví dụ như RSN_1 và RSN_2, trong Hình 2), mà ở mỗi sàn đồ thị liên kết nội bộ tạo thành một tô-pô *RSN* (như trình bày ở mục II, là kết hợp của lưới với tập các random link[5] tức liên kết ngẫu nhiên), trên đó có một số nút được chọn để tạo ra trục "bus" được gọi là bus-node (trình bày dưới đây). Các RSN_i được kết nối với nhau thông qua Bus-way, được tạo bởi danh sách các bus-node. Một gói tin di chuyển từ các nút thuộc RSN_1 đến các nút thuộc RSN_2 sẽ cần phải đi qua nhiều bus-node trên Bus-way. Để tạo ra kết nối chặt hơn giữa các sàn (từ đó hạ thấp đường kính đồ thị mạng), mô hình tô-pô này cho phép bổ sung các random link (liên kết ngẫu nhiên) giữa các bus-node trên Bus-way. Cấu trúc hoàn thiện của tô-pô được chúng tôi trình bày sau đây.

Một cách khái quát, Bus-RSN được biểu diễn như đồ thị $G = G_1 \cup G_2 \cup ... \cup G_m$, mỗi đồ thị $G_i = (V_i, E_i)$ là một tô-pô dạng RSN được gọi là zone thứ i, ký hiệu z_i, ứng với mỗi sàn – không gian liên tục biệt lập. Mỗi zone được chia thành k khối có kích thước giống nhau được gọi là *block*. Trong mỗi block, một nút được chọn làm nút đại diện cho các nút khác, được gọi là *bus-node*, nằm ở vị trí trung tâm block. Các bus-node được liên kết lần lượt với nhau để tạo nên Bus-way. Sau khi Bus-way được tạo ra, các random link có thể được tạo bổ sung giữa 2 bus-node bất kỳ.

[5]Random link (liên kết ngẫu nhiên) là các liên kết kết nối giữa 2 nút mạng ngẫu nhiên trong mạng liên kết



Hình 2. Mô hình tô-pô Bus-RSN



Hình 3. Mô hình chi tiết của Bus-RSN

Hình 3 minh họa Bus-RSN gồm 4 zone, mỗi zone được chia thành các block có kích thước bằng nhau. Các nút được chọn làm đại diện của các block được đánh số thứ tự từ 1 đến 12 và tạo thành Bus-way. Ngoài các bus-link, là các liên kết cơ bản giữa các bus-node, còn có các random link kết nối 2 nút bất kỳ trên bus-node. Quá trình tạo thành một tô-pô dạng Bus-RSN được mô tả dưới đây có kèm mã giả (pseudo-code) minh họa bởi Algo. 1.

Algo. 1 là thủ tục được minh họa trong Hình 4 để khởi tạo mạng liên kết Bus-RSN với M zone. Kích thước của các zone và số block của zone đó được lưu trữ tại mảng A[] và B[] tương ứng. Ví dụ zone-i có kích thước là A[i] và có số block là B[i]. Các tham số đầu vào đều là các số nguyên dương và là bội số của 2.

Quá trình tạo tô-pô Bus-RSN có thể được chia thành 4

Thuật toán 1: Thuật toán xây dựng mạng liên kết theo mô hình Bus-RSN

Đầu vào: int M, số lượng zone của đồ thị;
int A[],lưu trữ kích thước của M zone;
int B[],lưu trữ số block của M zone;
Đầu ra : Tô-pô Bus-RSN

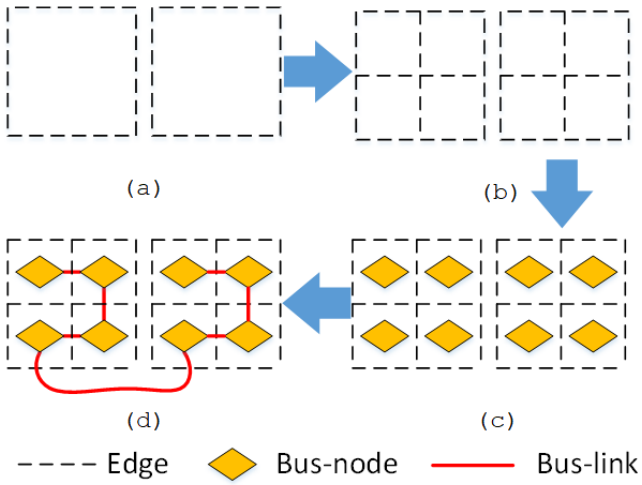
```
1  i = 0;
2  BusNodes ← ∅;
3  //khởi tạo danh sách bus-node của tô-pô
4  for (i = 0; i < M; i++){
5  |   Zone_i = GenerateRSN(A[i]);
6  |   Blocks = DivideBlock(Zone_i, B[i]);
7  |   BusNodeSelection(Blocks);
8  |   Update(BusNodes);
9  }
10 BuildBusWay(BusNodes);
```

giai đoạn như sau:

- *Khởi tạo tô-pô RSN* ($GenerateRSN(A[i])$): các tô-pô RSN với kích thước cho trước được khởi tạo bởi thủ tục tại dòng 4 của Algo. 1. RSN được tạo ra bằng cách thêm các random link (cho các cặp nút được lựa chọn ngẫu nhiên theo phân bố đồng nhất) của tô-pô cơ sở dạng lưới. Giai đoạn này tương ứng với Hình 4-a, RSN chưa được chia block và các bus-node chưa được chọn.
- *Phân chia tô-pô thành các khối*: Sau khi được khởi tạo tại giai đoạn 1, $Zone_i$ của mạng được chia thành B[i] block có kích thước lưới XxY bằng nhau. Các kích thước được chia sao cho X và Y là bội số của 2 và $|X-Y|$ là nhỏ nhất. Ví dụ: Zone có kích thước 128 có 4 block có kích thước là 4x8.
- *Lựa chọn bus-node*: Ngay sau khi zone được chia thành các block với kích thước giống nhau, các bus-node sẽ được tiến hành chọn và được lưu trữ vào mảng BusNodes[]. Các bus-node là nút ở trung tâm của block, được thực hiện bởi *BusNodeSelection()* như Hình 4-c. Nếu bus-node, là nút được chọn, có chứa random link với nút khác được tạo trong quá trình sinh RSN, link này sẽ bị xóa bỏ. Sau đó, các bus-node mới được thêm vào danh sách BusNodes qua thủ tục tại dòng 7.
- *Xây dựng Bus-way và tạo random link giữa các bus-node*: dựa trên danh sách BusNodes[], được thực hiện bởi thủ tục *BuildBusWay(BusNodes)* được minh họa ở Hình 4-d. Sau khi có được danh sách các bus-node từ các zone, các bus-node được liên kết lần lượt với nhau theo dạng tô-pô bus. Sau đó, các random link được thêm vào giữa 2 bus-node bất kỳ theo phân bố đồng nhất, nghĩa là các bus-node có bậc bằng nhau

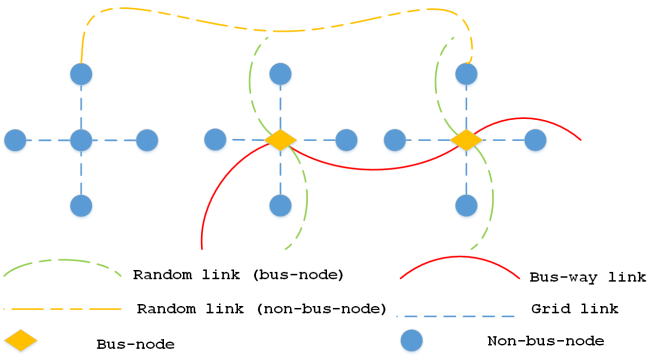
Nút và liên kết giữa các nút

Bus-RSN có 2 loại nút: các bus-node đã đề cập và các nút còn lại, gọi là non-bus-node. Do có cấu trúc cơ sở là



Hình 4. Quá trình RSN được chia thành các block, chọn bus-node và tạo Bus-way

tô-pô dạng lưới, một non-bus-node có 4 liên kết lưới tức grid-link giúp liên kết với 4 nút xung quanh, như Hình 5.



Hình 5. Chi tiết về Non-bus-node và Bus-node

Ngoài ra, non-bus-node còn có r-random link liên kết với các nút bất kỳ cùng zone với nó. Chúng tôi khuyến nghị sử dụng non-bus-node với $r = 2$ random link để bậc của các nút không tăng lên quá cao. Trong các tô-pô khuyến nghị như thế bậc của các non-bus-node là 6.

Trong phạm vi zone, bus-node cũng có thể được coi là một non-bus-node. Ngoài nhiệm vụ như một non-bus-node, bus-node giúp các zone liên kết với nhau, tạo sự liên thông trong đồ thị với 4 grid link, 2 random link. Tuy nhiên, 2 random link này chỉ kết nối tới các bus-node khác. Nếu một random link có 2 mút là 2 bus-node thuộc cùng một zone thì nó sẽ tham gia tích cực vào định tuyến nội vùng, ngược lại, nó giúp làm giảm đáng kể độ giải đường đi (hop-count) trong định tuyến liên vùng. Ngoài ra, bus-node còn có các bus-way link, dùng để liên kết các zone với nhau, tạo sự liên thông cho mạng và thực hiện các nhiệm vụ điều khiển, giám sát. Do vậy, bậc của bus-node là 8, như Hình 5.

Khuyến nghị sử dụng trong thực tiễn

Bus-RSN được đề xuất cho các DC cỡ nhỏ và vừa phù hợp mới xu hướng phát triển linh hoạt và tiết kiệm trong nước, nên với những điều kiện bối cảnh cụ thể đó chúng tôi cũng khuyến nghị là các ứng dụng thuê bao máy chủ trên DC nên được gán với nhóm các máy thuộc cùng một sàn. Điều này là khá phù hợp vì các khách hàng của DC loại này cũng thường có nhu cầu không quá lớn. Do đó liên kết giữa các sàn dù chỉ thực hiện hạn chế trên Bus-way cũng sẽ vẫn đảm bảo được nhu cầu truyền tin phục vụ cho quản lý, kiểm soát điều khiển hệ thống (vốn giao thông có tỷ trọng rất nhỏ so với giao thông dữ liệu ứng dụng). Việc sử dụng kiến trúc bus tối giản như thế giúp giảm thiểu chi phí liên kết giữa các sàn có thể ở khá xa nhau. Trong trường hợp có ứng dụng yêu cầu số lượng máy chủ tương đối nhiều và bắt buộc phải triển khai trên nhiều hơn 1 sàn, hệ thống sẽ có thể bổ sung các random link giữa các zone phù hợp sao cho đảm bảo yêu cầu mà không tốn kém (mục IV-2 sẽ trình bày chi tiết thông qua một case-study cụ thể). Đây có thể coi là ưu điểm chính của giải pháp tô-pô đề xuất nhờ vừa đảm bảo tiết kiệm chi phí thiết bị và cáp mạng vừa linh hoạt đáp ứng được yêu cầu cao hơn phát sinh.

2. Giải pháp định tuyến cho Bus-RSN

Algo. 1 giải quyết vấn đề sinh mạng kết nối Bus-RSN, tuy nhiên để có thể hoạt động được, cần có thuật toán định tuyến được giải quyết tại Algo. 2 là thủ tục tìm đường đi giữa 2 nút mạng bất kỳ của Bus-RSN. Đầu vào của thuật toán định tuyến là 2 nút mạng, đầu ra đường đi từ nút nguồn đến nút đích. Đường đi giữa 2 nút mạng là danh sách các nút mạng của tô-pô, được sắp xếp theo thứ tự gói tin đi qua từ nút nguồn đến nút đích.

Bảng II
ĐỊNH NGHĨA MỘT SỐ KÍ HIỆU ĐƯỢC SỬ DỤNG

Ký hiệu	Diễn giải
z_i	Định danh zone id của nút i
b_i	Định danh block id i
i	Node id của nút i
bn_i	Bus-node i
cbn_i	Bus-node gần nút i nhất
$nBlock$	Số block trong một đồ thị RSN

Với cách tổ chức kiến trúc Bus-RSN đã được trình bày, là mô hình hướng đến các DC được xây dựng trên không gian rời rạc (nhiều phòng, nhiều tầng khác nhau), nên trong mô hình này sử dụng các random link để thu nhỏ đường kính mạng. CORRA là thuật toán định tuyến ưu tiên sử dụng các random link. Vì vậy CORRA có thể tận dụng đặc điểm của mô hình mạng để có một DC có các tham số hiệu năng tốt. Các kết quả được thể hiện trong Phần V. Các ký hiệu chúng tôi sử dụng được chú thích trong Bảng II.

Địa chỉ của một nút trong Bus-RSN bao gồm 3 thành phần: định danh vùng $Zone_{id}(z_i)$, định danh của khối $Block_{id}(b_i)$, $Node_{id}(i)$, được cấu trúc thành $v(z_i,(b_i,i))$. Nút $v(1,(1,8))$ trong Hình 3 là nút thuộc z_1, b_1 và có i = 8. Tham số z_i của một nút được quyết định bởi zone quản lý nút đó, được dùng để phân biệt các nút thuộc zone nào. Ví dụ, với 2 nút u và v, nếu có $Zone_{id}$ giống nhau thì chúng cùng thuộc 1 zone, ngược lại, chúng thuộc 2 zone khác nhau. Trong Hình 3, hai nút $u,v \in z_1, t \in z_2$. Trong cùng một zone, cần phân biệt các block thuộc zone đó. Hai nút $u,s \in z_1$, nhưng $u \in b_1$ cho nên b_u là 1, còn $s \in b_3$ nên b_s là 3. Cuối cùng là $Node_{id}$ được dùng để phân biệt các nút trong cùng một block. Trong Hình 3, 2 nút u và v cùng thuộc b_1, z_1, có $Node_{id}$ lần lượt là 2 và 8. Như vậy, tọa độ của mỗi nút là riêng biệt, từ đó hỗ trợ quá trình định tuyến trên DC.

Trong mô hình Bus-RSN, có 2 trường hợp định tuyến: định tuyến nội vùng (intra-zone) và định tuyến liên vùng (inter-zone). Khi 2 nút nguồn đích ở trong cùng một zone (intra-zone), chúng tôi triển khai áp dụng thuật toán định tuyến CORRA [21]. CORRA là một thuật toán rút gọn (compact routing), nên bảng định tuyến của các nút sẽ không phải lưu quá nhiều thông tin. Ngoài ra, với đặc trưng của mình, CORRA tận dụng tối đa các random link như là các cầu nối giữa các vùng hàng xóm, từ đó cải thiện các tham số hiệu năng của mạng. Khi sử dụng thuật toán định tuyến này, non-bus-node sẽ thu thập thông tin các nút cùng với các random link trong vùng hàng xóm của nó là tập hợp các nút cách nút đang xét là 2 nút trung gian (hop) nếu di chuyển bằng grid-link.

Trong trường hợp thứ 2, khi cặp nút nguồn đích ở 2 zone khác nhau, gói tin cần đi qua bus-way. Bus-node có nhiệm vụ giúp các zone liên kết với nhau nên chúng cần lưu thông tin của các bus-node khác. Nói cách khác, bảng định tuyến của non-bus-node u cần lưu thông tin của các nút, random link trong vùng hàng xóm của u và các bus-node có trong mạng. Bảng định tuyến của bus-node cần lưu thông tin của các nút trong block mà nó đại diện cùng với các bus-node khác có trong mạng.

Hai trường hợp có thể được đặc tả như sau.

Định tuyến nội vùng: Intra-zone: là khả năng xảy ra khi hai nút cùng nằm trong một zone. Ví dụ, cặp nút nguồn $u,v \in z_1$ trong Hình 3. Chúng tôi sử dụng thuật toán định tuyến CORRA [21] khi định tuyến 2 nút trong cùng 1 zone để có thể sử dụng tối đa các random link, từ đó hiệu năng của mạng liên kết cũng được gia tăng.

Định tuyến liên vùng: Inter-zone: xảy ra khi cặp nút nguồn đích thuộc 2 zone khác nhau, ví dụ $S \in z_i$ và $T \in z_2$ trong Hình 3. Tương tự thuật toán định tuyến TZ [20], chúng tôi chia quá trình định tuyến cho trường hợp này thành 2 giai đoạn. Giai đoạn 1, gói tin từ nút nguồn được

định tuyến đến bus-node gần nút đích nhất. Giai đoạn 2, gói tin từ bus-node gần với nút đích nhất di chuyển đến nút đích.

Thuật toán 2: Thuật toán định tuyến cho mô hình Bus-RSN

Đầu vào: Bus-RSN;
source node $S(z_i,(b_i,i))$;
destination node $T(z_j,(b_j,j))$;
Đầu ra : Routing path from S to T

```
if (GetZoneId(S) = GetZoneId(T)){
    IntraZone(S,T);
}
else{
    Phase1(S, cbn_T);
    Phase2(cbn_T, T);
}
```

IV. ĐÁNH GIÁ BẰNG THỰC NGHIỆM

Để thuận tiện cho các độc giả chuyên ngành chúng tôi tiếp tục sử dụng các thuật ngữ kỹ thuật phổ biến trong tiếng Anh (có chú giải tiếng Việt trong xuất hiện lần đầu).

Bảng III
CÁC KÝ HIỆU TRONG CÁC HÌNH MINH HỌA THỰC NGHIỆM

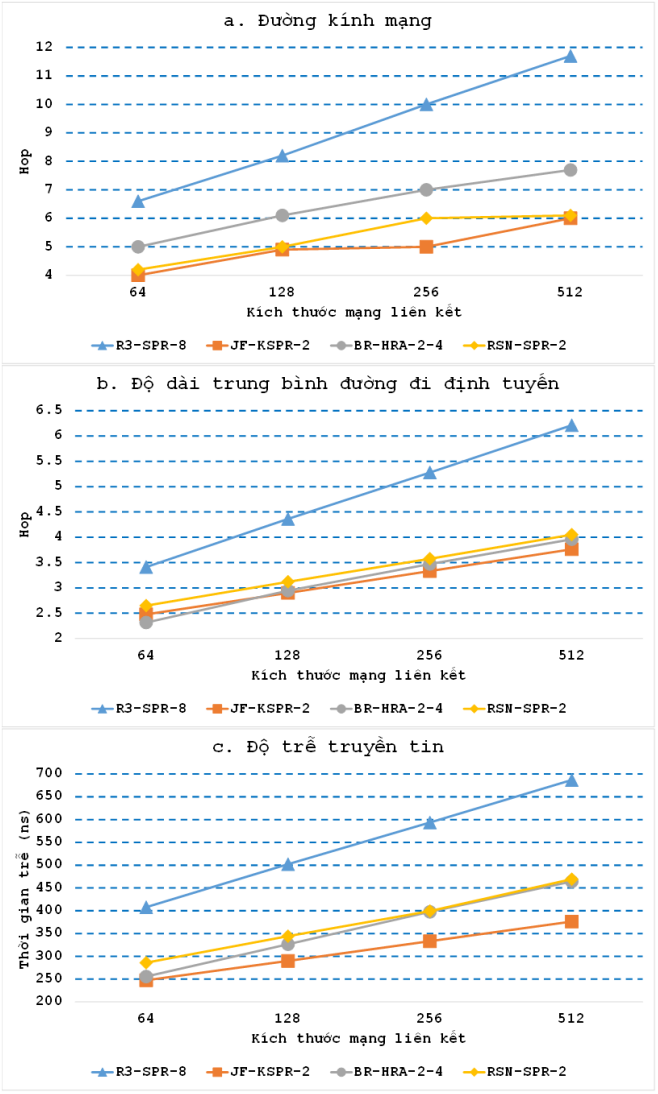
Tô-pô	Thuật toán định tuyến	Ký hiệu
Bus-RSN	Hybrid	BR-HRA-2-4
JellyFish	K-shortest path	JF-KSPR-2
R3	Shortest path	R3-SPR-2
RSN	Shortest path	RSN-SPR-2

Chúng tôi tiến hành khảo sát các tham số hiệu năng cơ bản của mạng liên kết, bao gồm Diameter (đường kính đồ thị), Average routing path length – ARPL (độ dài trung bình đường định tuyến), Average latency (độ trễ truyền tin trung bình giữa một cặp nút), Routing table size (kích thước bảng định tuyến), Total cable length (tổng độ cài cáp mạng), Network cost (chi phí thiết bị và cài đặt mạng), của tô-pô Bus-RSN cùng với các tô-pô quan trọng đã đề cập: JellyFish [2], RSN [8], R3 [3] thông qua công cụ phần mềm đánh giá có mô phỏng SSiNET [1].

Chúng tôi sử dụng và giải thích các kí hiệu được sử dụng trong các hình minh họa như tại Bảng III. Mỗi ký hiệu bao gồm tên tô-pô, thuật toán định tuyến tương ứng, số zone được sắp xếp. Riêng Bus-RSN có thêm tham số cuối là số lượng block trong mỗi zone, ví dụ 4 có nghĩa là mỗi zone được chia thành 4 khối. Tô-pô R3 [3] được tổ chức thành 8 zone để đảm bảo cấu trúc hyper-cube của nó và có số nút mạng tương đương với các tô-pô còn lại. Các mạng liên kết cũng được mô phỏng đặt vào các phòng (sàn) có địa điểm khác nhau, cách nhau 15m.

1. Đánh giá hiệu năng mạng liên kết

Trong phần này, chúng tôi trình bày về các tham số hiệu năng của các tô-pô sau khi được khảo sát theo 2 kịch bản.



Hình 6. Đánh giá tham số hiệu năng mạng bằng phân tích đồ thị theo kịch bản 1

Trong kịch bản thử nghiệm 1, các tô-pô được thực hiện khảo sát với cùng kích thước: 64, 128, 256, 512, được lắp đặt ở 2 zone cách nhau 15m. Các zone của tô-pô Bus-RSN được chia thành 4 block.

Hình 6-a là biểu đồ so sánh diameter giữa các mạng liên kết. Có thể thấy JellyFish có kết quả tốt nhất, khi ở cả 4 kích thước, JellyFish cho diameter thấp nhất, dao động từ 4 đến 6 hop, tiếp theo đó là RSN-SPR và Bus-RSN. Điều này có thể lý giải được bởi vì cấu trúc mạng của JellyFish là hoàn toàn ngẫu nhiên cùng thuật toán định tuyến *k-Shortest Path* nên diameter (Hình 6-a) và độ dài trung bình đường định tuyến (Hình 6-b) của JellyFish là thấp nhất, trong mọi

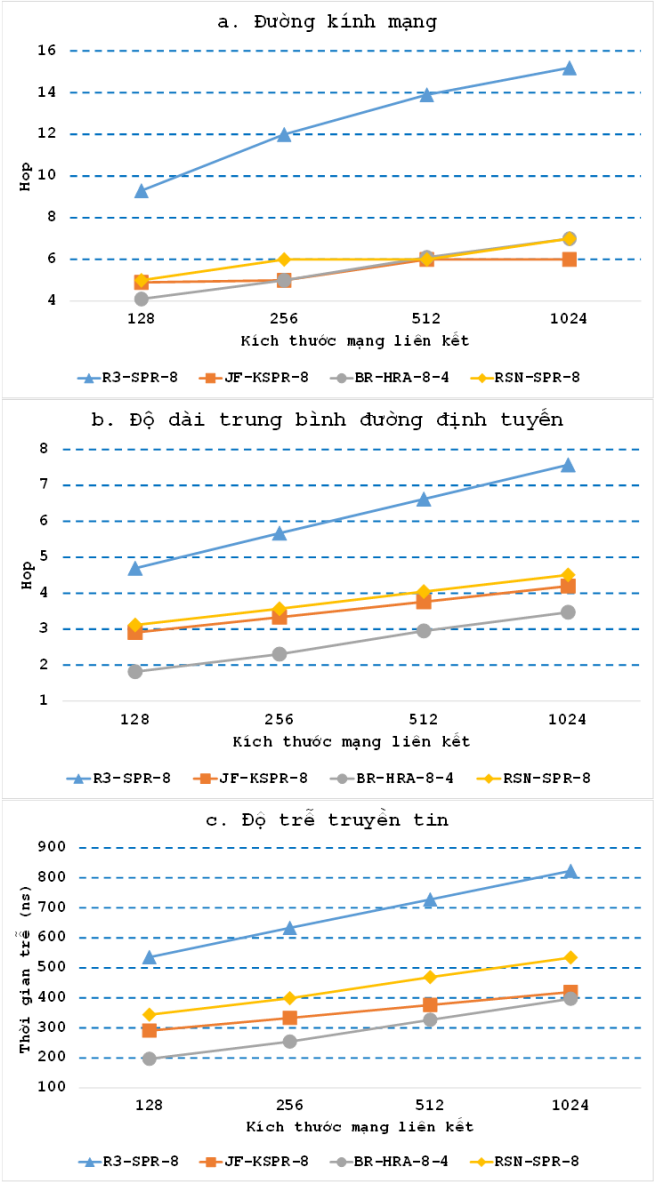
kích thước. Tô-pô RSN với kích thước nhỏ nên khi sử dụng thuật toán định tuyến SPR cho diameter khá nhỏ (xấp xỉ JellyFish). R3 cho diameter cao nhất vì tô-pô có cấu trúc hyber-cube, nên sẽ cần nhiều hop hơn khi gói tin di chuyển. Bus-RSN sử dụng thuật toán CORRA, thuật toán định tuyến ưu tiên sử dụng random link, nên đường kính mạng của nó cũng khá nhỏ, dao động trong khoảng 5,0 đến 7,7 khi kích thước mạng thay đổi từ 64 đến 512.

Do cấu trúc mạng JellyFish bao gồm hoàn toàn là liên kết ngẫu nhiên, ngoài ra nhờ sử dụng thuật toán định tuyến *k-Shortest Path*, nên JellyFish có độ dài trung bình đường định tuyến là thấp nhất trong 4 tô-pô. Bus-RSN và RSN cho kết quả xấp xỉ JellyFish ở các kích thước thử nghiệm khi kích thước mạng tăng từ 64 lên 512. Kết quả trong Hình 6-b cho thấy, đối với các kích thước được thử nghiệm, JellyFish không quá vượt trội so với Bus-RSN. Yếu tố chính khiến độ dài trung bình đường định tuyến của Bus-RSN không quá thua kém so với JellyFish đến từ kích thước của mạng kết nối.

Độ trễ trung bình (tính theo *ns*- nano second) phụ thuộc khá nhiều vào 2 yếu tố: đường kính mạng và độ dài trung bình đường định tuyến. Trong mô hình tính toán, độ trễ truyền tin được tính bằng tổng độ trễ tại các nút mạng (tính theo số hop-count) và tổng độ trễ trên quãng đường định tuyến (tính theo mét). Giá trị độ trễ thấp được xem là hiệu quả. Như đã trình bày trên về đường kính và độ dài trung bình đường định tuyến, nên JellyFish có độ trễ thấp nhất, tiếp đến là Bus-RSN. Ở kích thước 64, độ trễ của JellyFish khoảng 250 , trong khi đó Bus-RSN khoảng 256. Nhìn chung, Bus-RSN có độ trễ cao hơn Jellyfish, nhưng trung bình độ trễ toàn mạng là chấp nhận được, ví dụ Bus-RSN có độ trễ trung bình cao hơn 12,61% so với JellyFish ở kích thước mạng là 128. Kết quả độ trễ truyền tin trung bình giữa một cặp nút của các tô-pô được trình bày trong Hình 6-c.

Sau khi thực hiện khảo sát các tô-pô với kịch bản 1, chúng tôi tiếp tục khảo sát với kích thước lớn hơn: 128, 256, 512, 1024. Trong kịch bản này, các tô-pô được lắp đặt tại 8 zone, cách nhau đôi một 15m. Các zone của tô-pô Bus-RSN được chia thành 4 block.

Hình 7-a là kết quả chi tiết về đường kính mạng của các tô-pô. Ta có thể thấy rằng 3 tô-pô Bus-RSN, R3, RSN với các thuật toán định tuyến được sử dụng có đường kính mạng xấp xỉ nhau, dao động từ 4 đến 7. RSN và JellyFish sử dụng các thuật toán định tuyến mà bản chất là sử dụng đường đi ngắn nhất nên không bất ngờ khi 2 tô-pô này có đường kính mạng nhỏ. Bus-RSN sử dụng thuật toán CORRA, là thuật toán định tuyến ưu tiên sử dụng các random link. Với cấu trúc của Bus-RSN, các liên kết ngẫu nhiên xuất hiện ở mọi node trong tô-pô làm cho đường kính mạng của mạng nhỏ lại. R3 có đường kính mạng thấp vì các node mạng của tô-pô này thuộc nhiều zone khác nhau. Nên khi định tuyến, chúng phải đi qua nhiều nút biên trong R3), khiến cho đường kính mạng của R3 lớn hơn các tô-pô còn lại.



Hình 7. Đánh giá tham số hiệu năng mạng bằng phân tích đồ thị theo kịch bản 2

Khi chia mạng kết nối thành nhiều zone các nhau, Bus-RSN cho thấy lợi thế của mình với cấu trúc nhiều liên kết ngẫu nhiên và thuật toán ưu tiên sử dụng chúng. Điều này thể hiện trong hình 5, khi độ dài trung bình đường định tuyến của Bus-RSN thấp nhất trong 4 tô-pô, dao động trong khoảng từ 1,8 đến 2,5 với kích thước mạng từ 128 đến 1024. Trong điều kiện hạn chế cáp mạng, số lượng kết nối giảm xuống dẫn đến bậc của các nút mạng trong mạng JellyFish giảm xuống khiến mô hình này mất lợi thế. Độ dài trung bình đường định tuyến của JellyFish hơn Bus-RSN khoảng

60% ở kích thước 128 và 20% ở kích thước 1024. RSN có độ dài trung bình đường định tuyến xấp xỉ JellyFish. Do cấu trúc mạng đã được giải thích ở phần đường kính mạng, R3 có độ dài trung bình đường định tuyến thấp nhất trong các tô-pô với các kích thước được thử nghiệm.

Hình 7-c là kết quả thử nghiệm chi tiết độ trễ truyền tin trung bình giữa một cặp nút của các tô-pô. Độ trễ truyền tin trung bình giữa một cặp nút là thông số phụ thuộc vào đường kính mạng và độ dài trung bình đường đi định tuyến, nên không khó hiểu khi Bus-RSN có độ trễ truyền tin trung bình giữa một cặp nút tốt nhất trong các tô-pô với các kích thước thử nghiệm từ khoảng 200ns ở kích thước 128 đến 400ns ở kích thước 1024. Có thể thấy, với các kích thước nhỏ, Bus-RSN có lợi thế đối với JellyFish.

2. Đánh giá chi phí triển khai

Sau khi khảo sát các tham số hiệu năng, chi phí để xây dựng các DC theo các mô hình tô-pô trên được chúng tôi đánh giá theo 2 kịch bản đã được đưa ra.

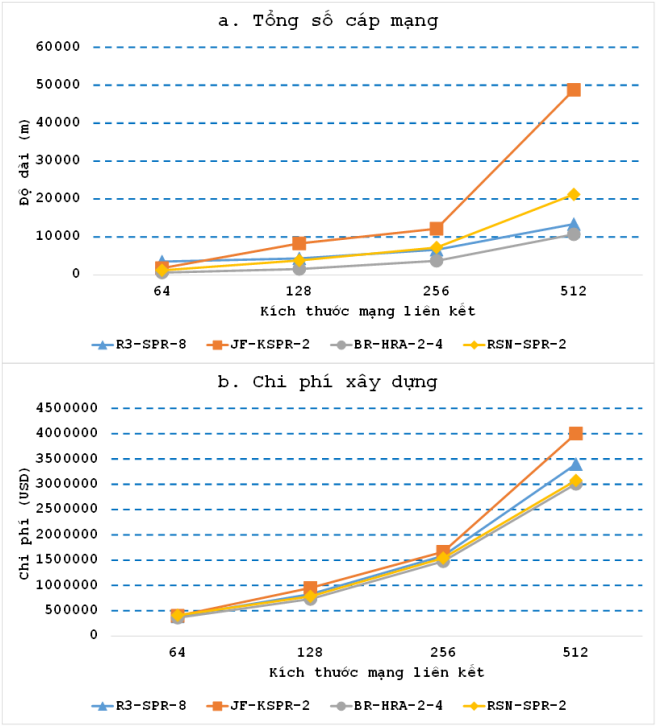
Với kịch bản 1, các tô-pô được thực hiện khảo sát với cùng kích thước: 64, 128, 256, 512, được lắp đặt ở 2 zone cách nhau 15m. Các zone của tô-pô Bus-RSN được chia thành 4 block.

Tổng chiều dài cáp (Total Cable length) là lợi thế đặc thù của Bus-RSN. Do cấu trúc của Bus-RSN là các zone tách biệt được kết nối thông qua 1 bus-way, cho nên khi DC được lắp đặt tại các phòng khác nhau sẽ chỉ tốn thêm chi cáp nối cho việc lắp đặt bus-way. Nhưng đối với JellyFish, khi xây dựng theo mô hình này, các liên kết ngẫu nhiên sẽ bị kéo dài khi DC phải lắp đặt tại các phòng (sàn) riêng biệt.

Kết quả trong Hình 8-a cho thấy rõ sự bất lợi về số lượng cáp nối khi mô hình JellyFish [2] phải lắp đặt tại các DC như vậy. Thử nghiệm cho thấy Bus-RSN có kết quả tốt nhất khi sử dụng ít cáp nối và thấp hơn JellyFish, thấp hơn 81,53% ở kích thước mạng 128 nút.

Chi phí đầu tư cho thiết bị mạng chiếm một phần quan trọng trong chi phí cài đặt mạng liên kết. Để đơn giản hóa mô hình tính toán, chúng tôi lựa chọn chi phí cho một thiết bị chuyển mạch ở mức 500$ cho mỗi cổng (port) dựa trên khảo sát trong [25]. Mỗi cổng tương ứng với một liên kết của thiết bị chuyển mạch đó trong mạng liên kết. Chi phí cài đặt mạng còn bao gồm cả chi phí cho các dây nối tức cáp mạng (cable) dùng để liên kết các thiết bị với nhau. Chi phí dành cho một kết nối bằng giá trị của dây nối đó (bao gồm chi phí dây nối, và chi phí đầu kết nối) cộng thêm 25% chi phí trung bình dành cho nhà sản xuất (chi phí sản xuất, phân phối, tiền lãi) và chi phí để lắp đặt thực tế (installation cost).

$$Cable\_cost = (Cable\_length * Cost\_per\_m + Connector\_Cost) * 1,25 + Installation\_Cost$$



Hình 8. Tổng chiều dài cáp và tổng chi phí triển khai kết nối theo kịch bản 1

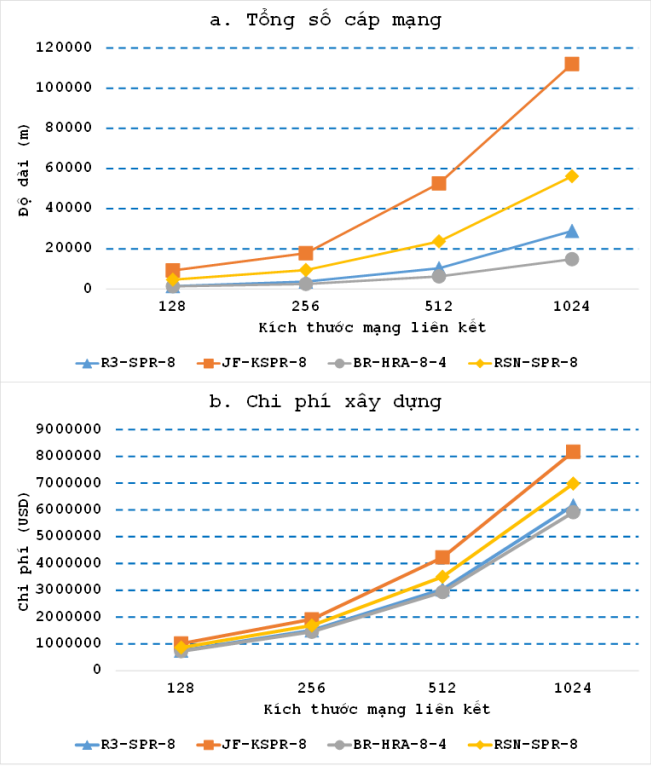
Chi phí trung bình cho việc cài đặt các kết nối giữa hai nút mạng bên trong một tủ mạng yêu cầu chi phí 2,5$ và 6,5$ đối với kết nối giữa hai nút mạng ở hai tủ mạng khác nhau. Chi phí cho mỗi mét cáp quang là $Cost\_per\_m = 10/m$, và mỗi đầu nối $Connector\_Cost = 188$\$. Vậy chi phí cho kết nối này được tính bằng $(10*5+188*2)*1,25+6,5 = 539$\$. Tổng chi phí để xây dựng DC được tính theo công thức:

$$Total\_Cost = Cable\_cost + switch\_port * 500 * network\_size$$

Sau đây chúng tôi thử khảo sát chi phí cho một số trường hợp ứng dụng có thể gặp trong thực tiễn (case studies). Đầu tiên ta khảo sát trường hợp như đã đề cập ở mục III.1 (phần khuyến nghị sử dụng), khi mà có ứng dụng khai thác yêu cầu sử dụng vượt quá số lượng máy chủ trong một phòng, ví dụ như cần tất cả các máy chủ tại vùng z_1 và một phần máy chủ trong vùng z_2. Trong trường hợp đó, Bus-RSN cung cấp khả năng linh hoạt trong việc kết nối giữa hai vùng xác định bằng cách thay đổi một số kết nối trong vùng z_1 tới một số nút mạng trong vùng z_2. Bằng cách

thay thế các thiết bị switch (chuyển mạch) tại các bus-node từ loại 8 cổng lên 12 cổng, làm gia tăng các cầu kết nối tới các bus-node khác trong vùng z_2. Đồng thời cấu hình lại địa chỉ của các nút được chọn trong vùng đó, sao cho tương đồng với các địa chỉ các nút trong vùng nhưng không cần thay đổi vị trí (địa lý) của các nút đó. Có thể thấy rõ là, việc thay thế switch cũng như chi phí cho cáp mạng so với tổng chi phí xây dựng DC là không đáng kể và các switch được thay thế có thể tận dụng lại trong việc mở rộng DC sau này. Ví dụ, yêu cầu tính toán cần zone z_1 và block b_5, b_6, thuộc zone z_2. Với tính linh hoạt của Bus-RSN, hiệu suất tính toán có thể tăng lên bằng cách thêm liên kết giữa các bus-node thuộc zone z_1 và bus-node của block b_5 và b_6. Như vậy, chúng ta chỉ cần thay 2 switch có 8 port, 1 switch tương ứng với bus-node thuộc zone $z-1$ và 1 switch tương ứng với bus-node của b_5 hoặc b_6, bằng loại switch 12 port. Với cấu hình hiện tại, mỗi bus-node khi liên kết đến zone khác cần 15m cáp mạng. Chi phí cho sự thay đổi này là: 12x500x2 + 15x10 = 12.150 (USD), so với chi phí xây dựng ban đầu là chỉ tăng thêm khoảng 2,2% – khá khiêm tốn!

Chúng tôi tiếp tục khảo sát chi phí với kịch bản 2.



Hình 9. Tổng chiều dài cáp và tổng chi phí triển khai kết nối theo kịch bản 2

Được thiết kế để phù hợp với các không gian phân biệt nên Bus-RSN sử dụng cáp mạng ít nhất trong 4 tô-pô mạng với các kích thước được thử nghiệm, 1344 m ở kích thước 128 và 14861 ở kích thước 1024. Với kích thước 128, JellyFish sử dụng nhiều hơn 5,89 lần số lượng cáp mạng mà Bus-RSN cần, con số này là 6,5 với kích thước 1024. Kết quả chi tiết được thể hiện trong Hình 9-a.

Hình 9-b là kết quả chi tiết khi khảo sát tổng chi phí cho các tô-pô mạng dựa trên kịch bản thí nghiệm. Trong cùng một điều kiện (chi phí lắp đặt, chi phí vật liệu...) Bus-RSN là tô-pô có chi phí thấp nhất. Do các mạng kết nối có cùng số lượng switch, Bus-RSN sử dụng ít cáp mạng nhất (Hình 7) nên không khó hiểu khi Bus-RSN có chi phi lắp đặt tiết kiệm nhất trong kịch bản thử nghiệm, khoảng 726.573 USD ở kích thước 64 và 5.914.163 USD ở kích thước 1024. Trong khi đó, chi phí cần cho mô hình JellyFish hơn 38,9% so với Bus-RSN.

Chúng tôi tiến hành khảo sát kỹ hơn, mô phỏng việc lắp đặt DC trên các sàn tương tự thực tế bằng 2 cấu hình. Cấu hình 1 với DC được triển khai trong 4 zone có khoảng cách đồng đều nhau và cách nhau 15 mét. Trong cấu hình 2, giả sử rằng, ban đầu DC được lắp đặt 2 zone z_1 và z_2 cạnh nhau, với khoảng cách cáp kết nối là 15 mét (cùng tầng, ví dụ tầng 1). Sau đó, DC được mở rộng với z_3 được lắp đặt ở tầng trên (ví dụ tầng 2), cách tầng 2 là 15m. Cuối cùng, DC tiếp tục được mở rộng với z_4, 15m cao hơn tầng 3.

Với thiết kế 4 zone, tổng chiều dài cáp của Bus-RSN thấp hơn nhiều chiều dài cáp của JellyFish, chỉ bằng 22,18% và 17,43% với kích thước 128 và 256 (Bảng 4), tương ứng. Với cấu hình 2, khoảng cách giữa các zone tăng lên đáng kể, thì tổng chiều dài cáp của Bus-RSN vẫn duy trì mức thấp hơn của Jellyfish, ví dụ chỉ chiếm 22,75% tại kích thước mạng 256.

Ngoài ra, chúng tôi thực nghiệm với 4 sàn có kích thước và khoảng cách khác nhau, các mặt sàn có kích thước z_1, z_2 và z_3 lần lượt là 32 nút và z_4 là 128, tổng nút toàn mạng là 224 nút. Kết quả thực nghiệm cho thấy, Bus-RSN sử dụng ít cáp mạng hơn 83,19% và đạt tổng chi phí thấp hơn 11,2% khi so sánh với JellyFish. Lý do chính là bởi số lượng cáp kết nối giữa các nút mạng liên vùng của tô-pô nhiều và tỏ ra bất lợi khi kết nối giữa các phòng máy chủ đặt cách xa nhau. Do vậy, mô hình Bus-RSN phù hợp với điều kiện triển khai thực tế khi các phòng lắp đặt máy chủ được bố trí ở xa nhau và có kích thước khác nhau.

V. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất mô hình tô-pô lai Bus-RSN nhằm giải quyết bài toán xây dựng DC của các doanh nghiệp nhỏ và vừa. Với đặc thù thiết kế nhắm đến tính kinh tế và linh hoạt, Bus-RSN có thể lắp đặt trong không gian gồm nhiều phòng/sàn phân biệt. Ngoài ra, chi phí đầu tư ban đầu để xây dựng DC theo mô hình này phù hợp với các doanh nghiệp có nguồn vốn hạn hẹp. Tính linh

Bảng IV
TỔNG CÁP MẠNG TRONG TRƯỜNG HỢP KHÁC NHAU VỀ KHOẢNG CÁCH GIỮA CÁC ZONE

Cấu hình 1 (15 mét đều)			
Tổng nút	BR-HRA-4-4	JF-kSPR-4	Tỉ lệ %
64	653,95	3.304,90	19,79%
128	1.734,40	7.819,10	22,18%
256	3.145,05	18.042,40	17,43%
512	7.404,80	46.776,40	15,83%
Cấu hình 2 (khoảng cách không đều)			
64	798,70	3.422,60	23,34%
128	1.870,25	8.220,60	22,75%
256	3.293,95	19.058,40	17,28%
512	7.563,20	49.048,10	15,42%

hoạt trong khả năng mở rộng giúp doanh nghiệp có thể mở rộng DC một cách dễ dàng.

Các kết quả thí nghiệm được chúng tôi tổng hợp và trình bày chi tiết tại phần IV. Với những kết quả đạt được, có thể thấy rằng, Bus-RSN là một mô hình mạng tiềm năng khi sở hữu các tham số hiệu năng tốt và chi phí lắp đặt khiêm tốn.

Trong kịch bản thử nghiệm 1, các tham số hiệu năng: đường kính mạng, trung bình độ dài đường định tuyến và độ trễ truyền tin trung bình giữa một cặp nút không phải là điểm mạnh Bus-RSN khi đạt giá trị kém hơn của JellyFish nhưng không phải là kém nhiều, ví dụ độ trễ kém 12,61% khi so với JellyFish ở kích thước mạng 128, nhưng độ trễ này là chấp nhận được trong các ứng dụng nhỏ, DC vừa và nhỏ.

Tuy nhiên, khi thực hiện kịch bản thử nghiệm 2, Bus-RSN cho thấy kết quả tốt khi các tham số hiệu năng và chi phí đều tốt nhất. Ví dụ: độ dài trung bình đường định tuyến của JellyFish hơn Bus-RSN khoảng 60% ở kích thước 128 và 20% ở kích thước 1024.

Trong cả 2 kịch bản thử nghiệm, tổng chiều dài cáp và chi chí thiết bị tiết kiệm của Bus-RSN đều tốt hơn so với các tô-pô trên. Trong kịch bản thử nghiệm 1, tổng trung bình cáp ít hơn 81,53% và nhờ thế tổng chi phí giảm 23,08% khi so với JellyFish ở kích thước mạng 128. Với kịch bản 2, JellyFish sử dụng nhiều cáp mạng hơn 5,89 lần số lượng cáp mạng mà Bus-RSN cần ở kích thước 128, con số này là 6,5 ở kích thước 1024.

Cấu trúc mạng có thể lắp đặt với switch thương mại cùng với số lượng cáp mạng nhỏ dẫn đến chi phí để đầu tư xây dựng DC theo mô hình Bus-RSN là khá tiết kiệm so với các mô hình khác. Như vậy, mô hình Bus-RSN tiềm năng, phù hợp với các DC cỡ vừa và nhỏ, mà trong đó không đòi hỏi tốc độ tính toán thật cao (chấp nhận độ trễ) nhưng đảm bảo chi phí lắp đặt thấp.

Trong phạm vi của bài báo này, mô hình Bus-RSN mới chỉ được phân tích và đánh giá thông qua các thông số hiệu năng cơ sở (yếu tố đồ thị) mà chưa được thử nghiệm trong môi trường giả lập thực tế. Trong tương lai, chúng tôi dự định sẽ đánh giá các yếu tố hiệu năng nâng cao như thông lượng và điện năng tiêu thụ của mạng để có thể đưa ra đánh giá đầy đủ nhất về mô hình Bus-RSN.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Quỹ Phát triển khoa học và công nghệ Quốc gia (NAFOSTED) trong đề tài mã số 102.02-2017.316.

TÀI LIỆU THAM KHẢO

[1] Kiều Thành Chung, Nguyễn Tiến Thành, Nguyễn Khanh Văn, *"Một tiếp cận thiết kế công cụ phần mềm đánh giá hiệu năng mạng liên kết kích thước lớn"*, Chuyên san Các công trình Nghiên cứu và Phát triển Công nghệ thông tin và Truyền thông, 2019

[2] Singla, Ankit and Hong, Chi-Yao and Popa, Lucian and Godfrey, P Brighten, *"Jellyfish: Networking data centers randomly"*, Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), 225-238, 2012

[3] Luo, Lailong and Guo, Deke and Li, Wenxin and Zhang, Tian and Xie, Junjie and Zhou, Xiaolei, *"Compound graph based hybrid data center topologies"*, Frontiers of Computer Science, 9, 6, 860-874, 2015

[4] Al-Fares, Mohammad and Loukissas, Alexander and Vahdat, Amin, *"A scalable, commodity data center network architecture"*, ACM SIGCOMM computer communication review, 38, 4, 63-74, 2008

[5] Lebiednik, Brian and Mangal, Aman and Tiwari, Niharika, *"A survey and evaluation of data center network topologies"*, arXiv preprint arXiv:1605.01701, 2016

[6] Guo, Deke and Chen, Tao and Li, Dan and Liu, Yunhao and Liu, Xue and Chen, Guihai, *"BCN: Expansible network structures for data centers using hierarchical compound graphs"*,2011 Proceedings IEEE INFOCOM, 61-65, 2011

[7] Guo, Chuanxiong and Wu, Haitao and Tan, Kun and Shi, Lei and Zhang, Yongguang and Lu, Songwu, *"Dcell: a scalable and fault-tolerant network structure for data centers"*, Proceedings of the ACM SIGCOMM 2008 conference on Data communication, 75–86, 2008

[8] Koibuchi, Michihiro and Matsutani, Hiroki and Amano, Hideharu and Hsu, D Frank and Casanova, Henri, *"A case for random shortcut topologies for HPC interconnects"*, 2012 39th Annual International Symposium on Computer Architecture (ISCA), 177–188, 2012

[9] Farrington, Nathan and Porter, George and Radhakrishnan, Sivasankar and Bazzaz, Hamid Hajabdolali and Subramanya, Vikram and Fainman, Yeshaiahu and Papen, George and Vahdat, Amin, *"Helios: a hybrid electrical/optical switch architecture for modular data centers"*, Proceedings of the ACM SIGCOMM 2010 conference, 339–350, 2010

[10] Guo, Chuanxiong and Lu, Guohan and Li, Dan and Wu, Haitao and Zhang, Xuan and Shi, Yunfeng and Tian, Chen and Zhang, Yongguang and Lu, Songwu, *"BCube: a high performance, server-centric network architecture for modular data centers"*, Proceedings of the ACM SIGCOMM 2009 conference on Data communication, 63–74, 2009

[11] Gyarmati, László and Trinh, Tuan Anh, *"Scafida: A scale-free network inspired data center architecture"*, ACM SIGCOMM Computer Communication Review, 40, 5, 4–12, 2010

[12] Wu, Haitao and Lu, Guohan and Li, Dan and Guo, Chuanxiong and Zhang, Yongguang, *"MDCube: a high performance network structure for modular data center interconnection"*, Proceedings of the 5th international conference on Emerging networking experiments and technologies, 25–36, 2009

[13] Greenberg, Albert and Hamilton, James R and Jain, Navendu and Kandula, Srikanth and Kim, Changhoon and Lahiri, Parantap and Maltz, David A and Patel, Parveen and Sengupta, Sudipta, *"VL2: a scalable and flexible data center network"*, Proceedings of the ACM SIGCOMM 2009 conference on Data communication, 51–62, 2009

[14] Shin, Ji-Yong and Wong, Bernard and Sirer, Emin Gün, *"Small-world datacenters"*, Proceedings of the 2nd ACM Symposium on Cloud Computing, 1–13, 2011

[15] Yu, Ye and Qian, Chen, *"Space shuffle: A scalable, flexible, and high-bandwidth data center network"*, 2014 IEEE 22nd International Conference on Network Protocols, 13–24, 2014

[16] Fujiwara, Ikki and Koibuchi, Michihiro and Matsutani, Hiroki and Casanova, Henri, *"Skywalk: A topology for HPC networks with low-delay switches"*, 2014 IEEE 28th International Parallel and Distributed Processing Symposium, 263–272, 2014

[17] Kim, John and Dally, Wiliam J and Scott, Steve and Abts, Dennis, *"Technology-driven, highly-scalable dragonfly topology"*, 2008 International Symposium on Computer Architecture, 77–88, 2008

[18] Niranjan Mysore, Radhika and Pamboris, Andreas and Farrington, Nathan and Huang, Nelson and Miri, Pardis and Radhakrishnan, Sivasankar and Subramanya, Vikram and Vahdat, Amin, *"Portland: a scalable fault-tolerant layer 2 data center network fabric"*, Proceedings of the ACM SIGCOMM 2009 conference on Data communication, 39–50, 2009

[19] Wang, Guohui and Andersen, David G and Kaminsky, Michael and Papagiannaki, Konstantina and Ng, TS Eugene and Kozuch, Michael and Ryan, Michael, *"c-Through: Part-time optics in data centers"*, Proceedings of the ACM SIGCOMM 2010 conference, 327–338, 2010

[20] Thorup, Mikkel and Zwick, Uri, *"Compact routing schemes"*, Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures, 1–10, 2001

[21] Thanh, Chung Kieu and The, Anh Mai and Bui, Cuong and Pham, Hai D and Nguyen, Khanh-Van, *"An efficient compact routing scheme for interconnection topologies based on the random model"*, Proceedings of the Eighth International Symposium on Information and Communication Technology, 189–196, 2017

[22] Dally, William James and Towles, Brian Patrick, *"Principles and practices of interconnection networks"*, 2004

[23] Kieu, Thanh-Chung and Nguyen, Khanh-Van and Truong, Nguyen T and Fujiwara, Ikki and Koibuchi, Michihiro, *"An interconnection network exploiting trade-off between routing table size and path length"*, 2016 Fourth International Symposium on Computing and Networking (CANDAR), 666–670, 2016

[24] Aljazzar, Husain and Leue, Stefan, *"K∗: A heuristic search algorithm for finding the k shortest paths"*, Artificial Intelligence, 175, 18, 2129–2154, 2011

[25] Mudigonda, Jayaram and Yalagandula, Praveen and Mogul, Jeffrey C, *"Taming the Flying Cable Monster: A Topology Design and Optimization Framework for Data-Center Networks."*, USENIX Annual Technical Conference, 2011

SƠ LƯỢC VỀ TÁC GIẢ

Kiều Thành Chung

Tốt nghiệp kỹ sư Công nghệ Thông tin năm 2003, tại Trường Đại học Bách Khoa Hà Nội (HUST), Thạc sĩ Công nghệ Thông tin năm 2010. Năm 2016, tác giả nghiên cứu tại Viện Công nghệ Thông tin Quốc Gia Nhật Bản (NII). Hiện nay là Nghiên cứu sinh tại Viện Công nghệ Thông tin và Truyền thông (SoICT)-HUST. Lĩnh vực nghiên cứu: mạng liên kết, thuật toán định tuyến.

Vũ Quang Sơn

Sinh ngày: 09/06/1999.
Hiện là sinh viên ngành công nghệ thông tin tại Học viện Công nghệ Bưu chính Viễn thông. Tác giả đang học tập và nghiên cứu tại SEDIC-LAB thuộc Trường Đại học Bách Khoa Hà Nội. Các lĩnh vực mà tác giả hiện đang nghiên cứu: mạng liên kết, thuật toán định tuyến và ứng dụng mô phỏng giả lập truyền tin.

Nguyễn Đăng Hải

Tốt nghiệp trường Đại học Bách khoa Hà Nội năm 1995; nhận bằng Thạc sĩ Tin học tại Viện tin học Pháp ngữ năm 1997; nhận học vị Tiến sĩ Khoa học Máy tinh của trường Thực hành Công nghệ cao – Cộng hòa Pháp năm 2010. Hiện là Giảng viên chính tại bộ môn Khoa học Máy tính, Viện Công nghệ thông tin và truyền thông, trường Đại học Bách khoa Hà Nội. Lĩnh vực nghiên cứu: tính toán hiệu năng cao, mô phỏng song song và phân tán, hệ điều hành nhúng.

Nguyễn Khanh Văn

Tốt nghiệp Kỹ sư Tin học tại Đại học Bách Khoa năm 1992, Thạc sỹ Khoa học Máy tính tại Đại học Wollongong (Úc) năm 2000, Tiến sĩ Khoa học Máy tính tại Đại học California-Davis (Mỹ) năm 2006. Hiện là Phó Giáo sư, giảng dạy và nghiên cứu tại Viện Công nghệ thông tin và truyền thông, Đại học Bách khoa Hà Nội. Lĩnh vực nghiên cứu: thuật toán và các mô hình lý thuyết trong tính toán phân tán và mạng máy tính (mạng liên kết, mạng cảm biến không dây), an toàn thông tin.

Phát hiện hoạt động bất thường của người bằng mạng học sâu tích chập kết hợp mạng bộ nhớ dài ngắn

Nguyễn Tuấn Linh, Nguyễn Văn Thủy, Phạm Văn Cường
Học viện Công nghệ Bưu chính Viễn thông
Tác giả liên hệ: Phạm Văn Cường, cuongpv@ptit.edu.vn
Ngày nhận bài: 17/04/2020, ngày sửa chữa: 24/05/2020
Định danh DOI: 10.32913/mic-ict-research-vn.v2020.n1.925

Tóm tắt: Bài báo này đề xuất một mô hình học sâu tích chập kết hợp với mạng bộ nhớ dài ngắn (CNN-LSTM) cho bài toán phát hiện các vận động bất thường của người sử dụng cảm biến đeo trên người. Nhờ tận dụng các đặc tính không-thời gian, kiến trúc đề xuất CNN-LSTM đã được thiết kế để tự động học và biểu diễn các đặc trưng hiệu quả trên dữ liệu cảm biến không thuần nhất. Kết quả thử nghiệm trên 4 tập dữ liệu được công bố cho thấy mô hình đề xuất đã cho kết quả cải tiến tốt hơn từ 2% đến 7% F1-score so với các mô hình học máy dựa trên trích xuất đặc trưng thủ công SVM, mô hình học sâu tích chập (CNN) và mô hình mạng bộ nhớ dài ngắn (LSTM).

Từ khóa: *cảm biến đeo, cảm biến gia tốc, mạng tích chập, mạng bộ nhớ dài ngắn.*

Title: **Human Abnormal Activity Detection with Deep Convolutional Long-Short Term Memory Networks**

Abstract: This work proposes Deep Convolutional Neural Long-Short Term Networks (CNN-LSTM) to address the problem of human abnormal activity detection using wearable sensors. Our proposed architecture effectively utilizes spatial-temporal characteristics of sensing data for automatically leanring and representing features from heterogeneous sensing data. Experimental results have demonstrated that the proposed method has improved from 2% to 7% F1-score better than several shallow and deep models including SVM, CNN and LSTM on 4 published datasets.

Keywords: *wearable Sensor, accelerometer, CNN, LSTM*

I. ĐẶT VẤN ĐỀ

Phát hiện vận động bất thường của con người là lĩnh vực nhận được nhiều sự quan tâm của cộng đồng nghiên cứu vì đây là lĩnh vực có nhiều ứng dụng trong thực tế như hỗ trợ cho người mất trí nhớ [1], theo dõi người bệnh đột quỵ [2], theo dõi chăm sóc người vận động bất thường [3]v.v... Vận động bất thường được xem là các hoạt động mà con người *không có chủ ý* và thường gây ra những hậu quả xấu đối với chủ thể. Một người bị *ngã* trong khi đang làm việc nhà hoặc một cú *trượt chân* do đường trơn trượt là các ví dụ về vận động bất thường. Những vận động bất thường này khi xảy ra sẽ gây nguy hiểm cho con người (đặc biệt là người cao tuổi). Trong những trường hợp như vậy, nếu có một hệ thống phát hiện và đưa ra những cảnh báo hoặc tự động kết nối đến người trợ giúp sẽ hạn chế được các rủi ro cũng như giảm thiểu các hậu quả do vận động bất thường đến người.

Hai phương pháp tiếp cận phổ biến để giải quyết bài toán vận động bất thường là: sử dụng cảm biến được tích hợp vào môi trường [6] và cảm biến đeo trên người [4, 5, 22]. Trong cách tiếp cận thứ nhất thì các cảm biến hình ảnh như camera số được thiết đặt để quan sát các hoạt động hàng ngày của người [7] hoặc cảm biến định danh (RFID) được gắn vào trong các vật dụng trong nhà để phát hiện người sử dụng những vật dụng nào, từ đó suy diễn ra các hoạt động hàng ngày và vận động bất thường của người mất trí nhớ tạm thời [1, 23]. Hạn chế của phương pháp sử dụng camera là có thể gây ra sự xâm lấn không gian riêng tư và việc phát hiện vận động bất thường thường bị giới hạn trong một phạm vi là vùng quan sát được của camera hoặc các cảm biến được tích hợp vào môi trường. Ngược lại, cách tiếp cận thứ hai bằng cảm biến đeo trên người thường không bị giới hạn bởi môi trường, đồng thời cũng giảm thiểu được viêc xâm lấn riêng tư. Hơn nữa, với sự phát triển nhanh chóng của các thiết bị điện tử kết nối Internet

vạn vật (Internet of Things) thì các thiết bị đeo ngày càng có sẵn trên thị trường với giá thành rẻ. Chính vì vậy trong nghiên cứu này chúng tôi tiếp cận bài toán phát hiện vận động bất thường theo cách tiếp cận dựa trên cảm biến đeo.

Thời gian gần đây, mặc dù lĩnh vực nghiên cứu này đang đạt được nhiều thành công, tuy nhiên vẫn còn nhiều thách thức cần phải giải quyết để có thể đưa được các hệ thống trên vào ứng dụng thực tế như: làm thế nào một hệ thống phát hiện được các vận động bất thường trong các ngữ cảnh thực tế khác nhau với độ chính xác cao để có thể sử dụng cho các ứng dụng cảnh báo. Trong khi đó, dữ liệu về vận động bất thường thường rất đa dạng, phức tạp và ít có sẵn do các vận động bất thường vô tình xảy ra trong khi thực hiện các hoạt động hàng ngày (bình thường). Điều này dẫn tới khó khăn khi huấn luyện mô hình học máy để đạt được độ chính xác đủ tốt cho việc phát hiện các vận động bất thường. Hơn thế nữa, dữ liệu về vận động bất thường thường mất cân bằng (Imbalanced) do tần suất của từng loại vận động bất thường khác nhau một cách tự nhiên.

Trong nghiên cứu này, chúng tôi đề xuất một mô hình mạng học sâu tích chập kết hợp với mạng bộ nhớ dài ngắn có khả năng học từ dữ liệu cảm biến không thuần nhất. Cụ thể hơn, có hai đóng góp chính trong nghiên cứu này:

- Chúng tôi đề xuất một phương pháp học bằng việc kết mô hình mạng học sâu tích chập (CNN) và mạng bộ nhớ dài ngắn để giải quyết bài toán phát hiện các vận động bất thường từ dữ liệu cảm biến không thuần nhất bao gồm cảm biến gia tốc, cảm biến con quay hồi chuyển và cảm biến từ tính. Trong đó, mô hình CNN đóng vai trò như bộ encoder được huấn luyện để học và biểu diễn các đặc trưng từ nhờ khai thác đặc tính không gian của dữ liệu cảm biến; còn mạng LSTM dùng đóng vai trò bộ suy diễn (decoder) tận dụng các đặc tính về thời gian của dữ liệu cảm biến.

- Chúng tôi đánh giá phương pháp đề xuất trên một số bộ dữ liệu đã được công bố rộng rãi. Kết quả cho thấy phương pháp đề xuất của chúng tôi hiệu quả hơn so với một số phương pháp truyền thống và phương pháp học sâu khác do chưa tận dụng được hai đặc tính không gian và thời gian của dữ liệu cảm biến.

Nghiên cứu của chúng tôi khác biệt với các nghiên cứu khác ở hai điểm chính. *Thứ nhất* là phương pháp đề xuất đã tận dụng kết hợp được các đặc tính về không-thời gian (Spatial-Temporal Features) từ dữ liệu cảm biến để khai thác việc học và biểu diễn đặc trưng hiệu quả. *Thứ hai* là mô hình đề xuất của chúng tôi chấp nhận đầu vào là dữ liệu cảm biến không thuần nhất đến từ các loại cảm biến khác nhau kết hợp lại để phát hiện các hoạt động bất thường.

II. CÁC NGHIÊN CỨU CÓ LIÊN QUAN

Phát hiện hoạt động bất thường đã và đang thu hút được sự quan tâm của cộng đồng nghiên cứu [11]. Trước đây, phương pháp tiếp cận phát hiện hoạt động bất thường chủ yếu dựa trên các mô hình học máy trong đó học có giám sát [12] được sử dụng phổ biến. Các dữ liệu (mẫu) được gán nhãn để các mô hình có thể học và mô hình được huấn luyện sẽ được đánh giá trên các dữ liệu mới. Do đó, trong trường hợp có các lớp hoạt động bình thường và bất thường, mô hình sẽ học các đặc tính của các điểm dữ liệu này và phân loại chúng là hoạt động bình thường hay bất thường. Bất kỳ điểm dữ liệu nào không phù hợp với lớp hoạt động bình thường sẽ được mô hình phân loại là bất thường [9].

Aran và đồng sự [4] đã đề xuất một phương pháp có thể tự động hoá quan sát và mô hình hoá hoạt động hằng ngày của người cao tuổi, qua đó giúp phát hiện hoạt động bất thường từ dữ liệu thu được bằng cảm biến. Trong phương pháp của họ, sự bất thường liên quan đến các vấn đề về tín hiệu sức khoẻ. Với mục đích này, họ đã tạo ra một mô hình không gian xác suất theo thời gian để có thể tóm lược toàn bộ các hoạt động hằng ngày. Họ định nghĩa sự bất thường là những thay đổi đáng kể từ những hoạt động đã được học và được phát hiện, hiệu suất phát hiện được đánh giá bằng phương pháp entropy chéo. Trong nghiên cứu của họ, khi một hoạt động bất thường được phát hiện, ngay lập tức sẽ có thông báo được gửi đến người chăm sóc.

Ordonez và đồng sự [14] đã thực hiện một phương pháp phát hiện bất thường dựa trên thống kê Bayes, từ đó giúp phát hiện hoạt động bất thường của con người. Phương pháp của họ có khả năng tự động hỗ trợ người già, người khuyết tật sống một mình bằng cách học và dự đoán các hoạt động tiêu chuẩn qua đó cải thiện hiệu suất của hệ thống chăm sóc sức khoẻ. Thống kê Bayes được sử dụng để phân tích dữ liệu thu thập được, dự đoán hoạt động dựa trên ba đặc trưng xác suất, bao gồm: xác suất kích hoạt cảm biến (Sensor Activation Likelihood), chuỗi cảm biến (Sensor Sequence Likelihood) và sự kiện cảm biến (Sensor Event Duration Likelihood).

Yahaya và đồng sự [11] đề xuất thuật toán phát hiện đặc trưng mới có tên máy vectơ hỗ trợ một lớp (One-class SVM) giúp phát hiện hoạt động bất thường từ các hoạt động bình thường diễn ra hằng ngày. Sự bất thường trong kiểu nằm ngủ có thể được coi là dấu hiệu của Sự suy giảm nhận thức nhẹ ở người cao tuổi hoặc các vấn đề liên quan đến sức khoẻ khác. Palaniappan và đồng sự [15] lại đặc biệt quan tâm đến các hoạt động bất thường ở người bằng cách loại trừ tất cả các hoạt động được coi là bình thường. Các tác giả định nghĩa hoạt động bất thường là các hoạt động bất ngờ xảy ra theo một cách ngẫu nhiên. Phương pháp SVM đa lớp được họ sử dụng làm trình phân loại để xác định các hoạt động dưới dạng bảng chuyển trạng thái. Điều này sẽ giúp trình phân loại tránh được các trạng thái không thể đưa ra được (không thể truy cập được) từ trạng thái hiện tại.

Hùng và đồng sự [16] đã đề xuất một phương pháp mới kết hợp SVM và HMM sử dụng một hệ thống các cảm biến thiết lập trong nhà (homecare sensory system). Mạng cảm biến RFID được sử dụng để thu thập các hoạt động hằng ngày của người cao tuổi. Mô hình Markov ẩn (HMM) được sử dụng để học từ dữ liệu được thu thập, trong khi SVM được sử dụng để ước tính liệu hoạt động đó của người cao tuổi có là hoạt động bất thường hay không. Bouchachia và đồng sự [17] lại đề xuất một mô hình RNN để giải quyết các vấn đề về nhận biết hoạt động và phát hiện hoạt động bất thường cho người cao tuổi bị chứng mất trí nhớ.

Mặc dù có một số nghiên cứu phát hiện hoạt động bất thường, tuy nhiên từ các nghiên cứu ở trên vẫn tồn tại một số điểm hạn chế như: Độ chính xác dự đoán hoạt động bất thường của các phương pháp học nông phụ thuộc khá nhiều kinh nghiệm trích chọn các đặc trưng theo kinh nghiệm chuyên gia. Trong khi đó, một số phương pháp học sâu lại chưa tận dụng đầy đủ đặc tính không-thời gian của dữ liệu cảm biến (đặc biệt là dữ liệu cảm biến không thuần nhất) mà nghiên cứu này tập trung giải quyết.

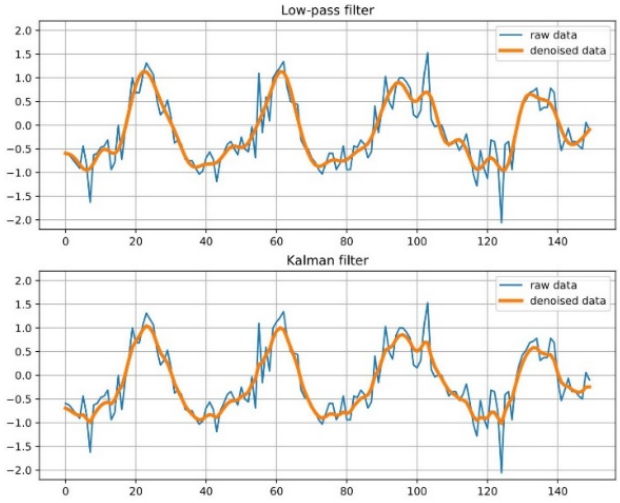
III. MÔ HÌNH MẠNG HỌC SÂU TÍCH CHẬP KẾT HỢP MẠNG BỘ NHỚ DÀI NGẮN (CNN-LSTM) CHO PHÁT HIỆN VẬN ĐỘNG BẤT THƯỜNG

Kiến trúc mạng học sâu tích chập kết hợp mạng bộ nhớ dài ngắn (CNN-LSTM) đề xuất được trình bầy trong Hình 2. Dữ liệu cảm biến được tiền xử lý trước khi đưa vào mạng. Kiến trúc mạng bao gồm 3 thành phần chính: tích chập, bộ nhớ dài ngắn và lớp đầu ra. Chi tiết được mô tả dưới đây.

Giả sử $S = S_k, k \in 1, .., 3$ tương ứng với 3 loại cảm biến: gia tốc, con quay hồi chuyển, và từ trường. Với cảm biến S_k, nó tạo ra một phép đo theo thời gian, các phép đo có thể được biểu thị bằng đối với ma trận V cho các giá trị đo với $n(k)$ là chiều của vectơ u cho các dấu thời gian (time stamps), $d(k)$ là kích thước cho mỗi phép đo (ví dụ: các phép đo dọc theo trục x, y, z đối với cảm biến), $n(k)$ là số phép đo. Chúng tôi chia các phép đo đầu vào V và u theo thời gian (các cột cho V) để tạo ra một chuỗi các chu kỳ thời gian không chồng lấn với chiều rộng $\tau, W = (V_t^{(k)}, u_t^{(k)})$ trong đó $|W| = T; \tau$ có thể khác nhau đối với các chu kỳ thời gian khác nhau. Để đơn giản chúng tôi giả sử chu kỳ thời gian là cố định. Sau đó, chúng tôi áp dụng biến đổi Fourier cho từng phần tử trongW bởi miền tần số chứa các tần số mẫu cục bộ tốt hơn, độc lập với cách tổ chức dữ liệu chuỗi thời gian trong miền thời gian. Chúng tôi tiến hành sắp xếp các đầu ra thành một $d(k) \times 2f \times T$ tensor $X^{(k)}$ trong đó f là thứ nguyên của miền tần số chứa các cặp pha và tần số cường độ f. Tập hợp các thang đo kết quả cho mỗi cảm biến $X = X^{(k)}$ sẽ là đầu vào của mô hình CNN-LSTM.

1. Lọc và tiền xử lý tín hiệu

Loại bỏ nhiễu: Tín hiệu cảm biến thường chứa nhiều tín hiệu nhiễu, điều này là do môi trường xung quanh có nhiều vật thể làm bằng kim loại hoặc do bản thân tự cảm biến sinh ra nhiễu. Vì vậy, các tín hiệu thu được cần phải thực hiện lọc bỏ nhiễu. Trong nghiên cứu này, chúng tôi sử dụng bộ lọc thông thấp và bộ lọc Kalman (như minh hoạ trong Hình 1).



Hình 1. Bộ lọc thông thấp (Low-pass filter) và bộ lọc Kalman.

Đây là những bộ lọc đơn giản, không đòi hỏi quá nhiều tài nguyên tín toán nhưng lại mang hiệu quả cao. Để tránh việc trễ, mỗi chuỗi dữ liệu được đưa qua bộ lọc hai lần, một lần theo chiều thuận và một lần ngược lại.

Tiếp đến chúng tôi căn chỉnh, phân chia các phép đo cảm biến và áp dụng biến đổi Fourier cho mỗi khối cảm biến. Đối với mỗi cảm biến, chúng tôi xếp các đầu ra miền tần số này thành $d(k) \times 2f \times T$ tenxor $X^{(k)}$, trong đó $d(k)$ là kích thước đo chiều cảm biến, f là kích thước miền tần số và T là số chu kỳ thời gian.

2. Thành phần mạng tích chập (CNN)

Các lớp tích chập có thể được chia làm hai phần: một mạng con tích chập riêng cho mỗi tensor cảm biến đầu vào $X^{(k)}$ và một mạng con tích chập gộp duy nhất cho đầu ra của K các mạng con tích chập riêng lẻ.

Do cấu trúc của mạng con tích chập riêng cho các cảm biến khác nhau là như nhau nên chúng tôi tập trung vào một mạng con tích chập riêng lẻ với đầu vào $X^{(k)}$. Cần lưu ý rằng $X^{(k)}$ là một $d^{(k)} \times 2f \times T$ tensor, trong đó $d^{(k)}$ cho biết kích thước chiều cảm biến, f là kích thước của miền tần số và T là số lượng chu kỳ thời gian. Đối với mỗi chu kỳ thời gian t, ma trận $X_{..t}^{(k)}$ sẽ được đưa vào kiến trúc CNN với ba lớp tích chập. Đặc trưng miền tần số và kích thước

số chiều được nhúng trong $X_{..t}^{(k)}$. Miền tần số thường chứa rất nhiều mẫu cục bộ ở một số tần số lân cận. Sự tương tác giữa các phép đo cảm biến thường bao gồm tất cả số chiều. Vì vậy, trước tiên, chúng tôi áp dụng các bộ lọc $2d$ có dạng $(d^{(k)}, cov1)$ cho $X_{..t}^{(k)}$ để học được sự tương tác giữa kích thước số chiều cảm biến và các mẫu cục bộ trong miền tần số với đầu ra $X_{..t}^{(k,1)}$. Tiếp theo, chúng tôi áp dụng các bộ lọc $1d$ với dạng $(1, cov2)$ và $(1, cov3)$ theo thứ bậc để tìm hiểu các mối quan hệ cấp cao hơn của $X_{..t}^{(k,2)}$ và $X_{..t}^{(k,3)}$.

Sau đó, chúng tôi tiến hành làm phẳng ma trận $X_{..t}^{(k,3)}$ thành vectơ $x_{..t}^{(k,3)}$ và ghép tất cả K vectơ $x_{..t}^{(k,3)}$ thành một K dòng ma trận $X_{..t}^{(3)}$ (là đầu vào của mạng con tích chập hợp nhất). Kiến trúc của mạng con tích chập hợp nhất tương tự như mạng con tích chập riêng lẻ. Bộ lọc $2d$ được chúng tôi sử dụng với $(K, cov4)$ để học các tương tác giữa các cảm biến K với đầu ra $X_{..t}^{(4)}$, sau đó bộ lọc $1d$ với $(1, cov5)$ và $(1, cov6)$ được áp dụng ở mức độ nâng cao hơn trên $X_{..t}^{(5)}$, $X_{..t}^{(6)}$.

Đối với mỗi lớp tích chập, CNN-LSTM học với 64 bộ lọc và sử dụng ReLU làm hàm kích hoạt. Ngoài ra, việc chuẩn hoá theo lô (batch) được áp dụng để mỗi lớp giảm sự thay đổi đồng biến nội bộ. Chúng tôi tiến hành làm phẳng đầu ra cuối cùng $X_{..t}^{(6)}$ thành vectơ $x_{..t}^{(6)}$. Ghép nối và chiều rộng chu kỳ thời gian $[\tau]$ thành $x_t^{(c)}$ làm đầu vào của các lớp LSTM.

3. Thành phần mạng bộ nhớ dài ngắn (LSTM)

Mạng thần kinh hồi qui (Recurrent Neural Networks-RNN) là những kiến trúc mạnh mẽ có thể giúp tính gần đúng và học các đặc trưng có ý nghĩa trong các chuỗi. Một biến thể của RNN là LSTM có thể lưu trữ được sự phụ thuộc dài hạn giữa các trạng thái (Long-term Dependencies). Trong mô hình đề xuất chúng tôi sử dụng cấu trúc tế bào (cell) xếp chồng lên nhau theo chiều chứa luồng thời gian từ đầu đến cuối (Start to End) của chuỗi dữ liệu thời gian (Time Series). Cấu trúc xếp chồng có thể chạy tăng dần khi có một chu kỳ thời gian mới, giúp xử lý luồng dữ liệu nhanh hơn. Đồng thời chúng tôi áp dụng dropout cho các kết nối giữa các lớp để chuẩn hoá và áp dụng chuẩn hóa theo bó hồi qui (Recurrent Batch Normalization) để giảm sự thay đổi đồng biến nội bộ giữa các bước thời gian (time steps). Đầu vào $x_t^{(c)}t$ với $t = 1, ..., T$ từ những lớp chập trước đó được đưa vào LSTM xếp chồng và tạo đầu ra $x_t^{(r)}$ với $t = 1, ..., T$ làm đầu vào của lớp đầu ra cuối cùng.

4. Lớp đầu ra

Đầu ra của lớp hồi qui là một chuỗi các vectơ $x_t^{(r)}$ với $t = 1, ..., T$. Đối với tác vụ định hướng hồi quy (regression-oriented), giá trị của mỗi phần tử trong vectơ $x_t^{(r)}$ nằm trong ± 1, $x_t^{(r)}$ mã hoá các đại lượng vật lý tại cuối chu kỳ thời gian t. Trong lớp đầu ra, chúng tôi muốn học một từ điển $\boldsymbol{W}_{out}$ (dictionary) Wout với một $\boldsymbol{b}_{out}$ bout (bias) để giải mã $x_t^{(r)}$ thành $\hat{y}_t$ sao cho $\hat{y}_t = \boldsymbol{W}_{out}.x_t^{(r)} + \boldsymbol{b}_{out}$. Do đó, lớp đầu ra là một lớp được kết nối đầy đủ trên đỉnh mỗi chu kỳ với chia sẻ tham số $\boldsymbol{W}_{out}$ và $\boldsymbol{b}_{out}$.

Đối với tác vụ phân loại, $x_t^{(r)}$ là vectơ đặc trưng tại khoảng thời gian t. Trước tiên, lớp đầu ra cần kết hợp $x_t^{(r)}$ thành một vectơ đặc trưng cố định để xử lý thêm. Đặc trưng trung bình theo thời gian là một lựa chọn. Các phương pháp nâng cao hơn có thể được áp dụng để tạo ra đặc trưng cuối cùng, ví dụ như mô hình chú ý (attention model) đã minh hoạ một cách có hiệu quả những tác vụ học quan trọng gần đây. Mô hình chú ý có thể được xem như là việc tính trung bình của các đặc trưng theo thời gian nhưng các trọng số được học bởi các mạng LSTM thông qua ngữ cảnh. Trong nghiên cứu này, chúng tôi vẫn sử dụng các đặc trưng trung bình theo thời gian để tạo ra các đặc trưng cuối cùng $x^r = (\sum_{t=1}^{T} x_t^{(r)})/T$. Sau đó, chúng tôi đưa x(r) và một lớp softmax để tạo ra các loại xác xuất dự đoán

IV. THỬ NGHIỆM

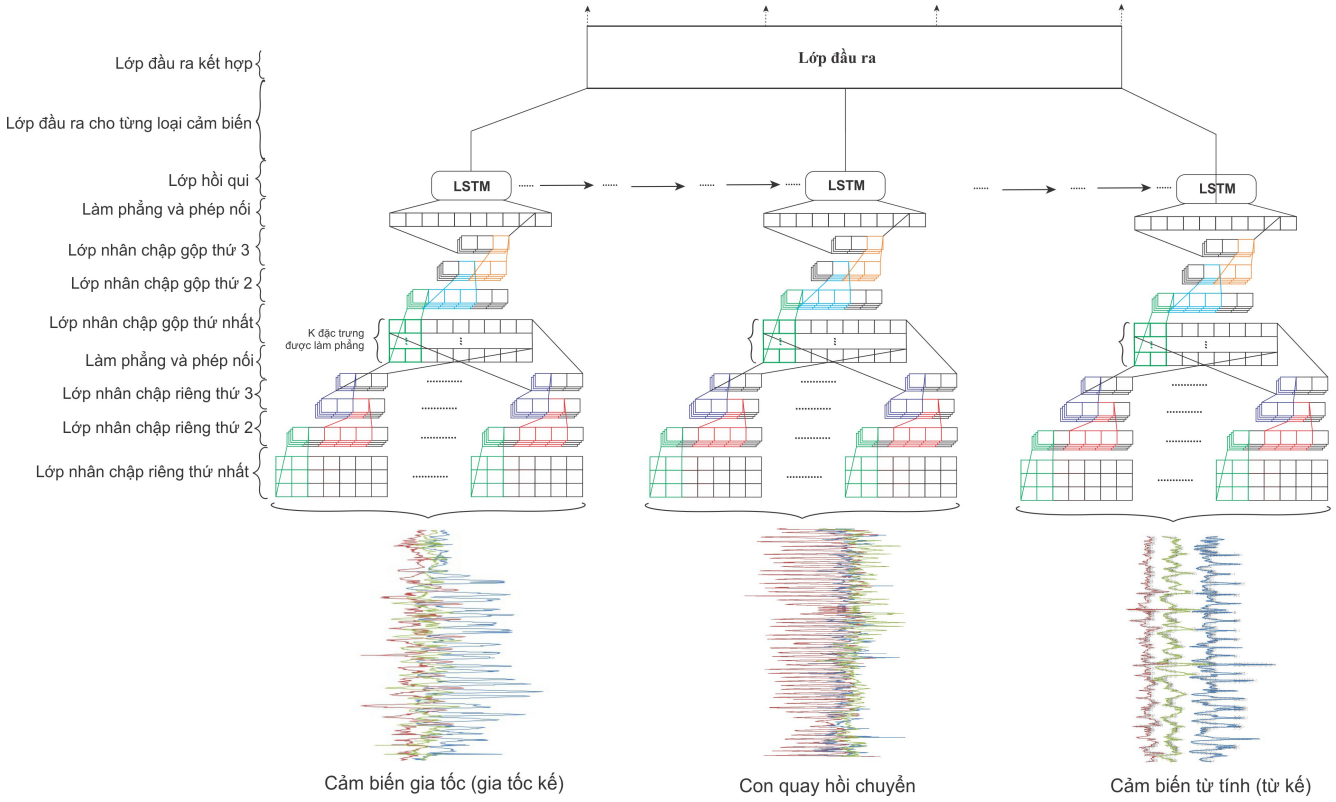
1. Tập dữ liệu

Chúng tôi sử dụng 4 tập dữ liệu, bao gồm UTD [14], MobiFall [15], PTITAct [9] và CMDFALL [8]. Chi tiết về mỗi tập dữ liệu như sau:

- *UTD* [14]: đây là tập dữ liệu được thu thập từ 12 người đeo 2 loại cảm biến là cảm biến gia tốc và con quay hồi chuyển với tần số lấy mẫu là 200Hz. Tập dữ liệu bao gồm 6 hoạt động bình thường và 1 hoạt động bất thường (ngã). Để huấn huyện mô hình CNN-LSTM với bộ dữ liệu này chúng tôi đóng băng (frozen) thành phần dành cho cảm biến từ tính và giảm tần số mẫu (downsampling) xuống còn 100 Hz;

- *MobiFall* [15]: là tập dữ liệu được thu thập từ 15 người để điện thoại thông minh trong túi quần. Dữ liệu cảm biến bao gồm cảm biến gia tốc và con quay hồi chuyển được thu thập với tần số lấy mẫu là 90Hz. Tập dữ liệu bao gồm 9 hoạt động bình thường và 4 hoạt động bất thường là các tư thế ngã khác nhau. Để huấn huyện mô hình CNN-LSTM với bộ dữ liệu này chúng tôi đóng băng (frozen) thành phần dành cho cảm biến từ tính và tái tạo tần số lấy mẫu (upsampling) lên 100 Hz bằng phương pháp GAN cho dữ liệu chuỗi thời gian [18];

- *PTITAct* [9]: là tập dữ liệu được thu thập từ 26 người gắn thiết bị internet vạn vật kết nối (IoT) ở thắt lưng. Thiết bị được tích hợp cảm biến gia tốc, con quay hồi chuyển, và từ kế. Dữ liệu cảm biến được thu thập với tần số lấy mẫu là 50Hz. Tập dữ liệu bao gồm 8 loại vận động bất thường (ngã ở các tư thế khác nhau) và 8 hoạt động bình thường. Trước khi huấn huyện mô hình CNN-LSTM, dữ liệu được



Hình 2. Kiến trúc mạng học sâu tích chập kết hợp mạng bộ nhớ dài ngắn (CNN-LSTM)

upsampling mẫu dữ liệu lên 100 Hz bằng phương pháp GAN cho dữ liệu chuỗi thời gian [18];

- *CMDFALL* [8]: là tập dữ liệu khá lớn được thu thập từ 50 người đeo 2 cảm biến tại vị trí cổ tay và thắt lưng. Tập dữ liệu gồm 9 nhãn hoạt động bình thường (như đi lại, nằm lên giường, ngồi xuống ghế v.v..) và 11 vận động bất thường (như ngã ngửa, ngã bên trái, đi loạng choạng, trượt chân ...) khác nhau. Do tần số lấy mẫu của tập dữ liệu là 50Hz nên khi thực nghiệm trên tập này, tập dữ liệu được upsampling mẫu dữ liệu lên 100 Hz bằng phương pháp GAN cho dữ liệu chuỗi thời gian [18]; Đây là những tập dữ liệu đã được công bố và được sử dụng khá rộng rãi trong cộng đồng nghiên cứu về phát hiện người ngã và vận động bất thường. Các tập dữ liệu đều có những thử thách như không cân bằng (imbalanced) và có nhiều vận động bất thường khá giống với các hoạt động thường ngày (ngã ra giường vs. ngồi và nằm xuống giường).

2. Độ đo đánh giá

Trong nghiên cứu này, chúng tôi sử dụng 3 độ đo là: *độ chính xác (precision), độ bao phủ (recall)* và *điểm cân bằng giữa độ chính xác và độ bao phủ* ($F1_{score}$):

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1_{score} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (3)$$

Trong đó, *True Positive* (TP) là tỉ lệ đo số lần mô hình phát hiện đúng vận động bất thường a và số lần thực tế xảy ra vận động bất thường a; ví dụ *ngã nghiêng bên phải* được phát hiện *đúng* là *ngã nghiêng bên phải*. *True Negative* (TN) là tỉ lệ đo số lần mô hình phát hiện đúng không phải vận động bất thường a và số lần thực tế xảy ra không phải vận động bất thường a; ví dụ *không phải là ngã nghiêng bên phải* được phát hiện *đúng không là ngã nghiêng bên phải*. *False Positive* (FP) là tỉ lệ đo số lần mô hình phát hiện sai vận động bất thường a và số lần thực tế xảy ra không phải vận động bất thường a; ví dụ *ngã nghiêng bên phải* được phát hiện *sai không phải là ngã nghiêng bên phải*. *False Negative* (FP) là tỉ lệ đo số lần mô hình phát hiện sai không phải vận động bất thường a và số lần thực tế xảy ra vận động bất thường a; ví dụ *không phải ngã nghiêng bên phải* được phát hiện *sai* là *ngã nghiêng bên phải*.

3. Các mô hình thử nghiệm (Baselines)

Chúng tôi thực nghiệm với một số mô hình sau:

- *Máy véc tơ hỗ trợ (SVM)*: với các bước tiền xử lý và trích xuất đặc trưng từ dữ liệu cảm biến được tham khảo

từ nghiên cứu [9]. Các véc tơ được tính từ các cửa sổ trượt được dùng để huấn luyện mô hình SVM với tham số C = 1, lămda là kết quả của tìm kiếm lưới (grid search) và hàm tích RBF.

- *Mạng CNN* [11]: được hiệu chỉnh để thích hợp với dữ liệu cảm biến [5] của từng tập dữ liệu thử nghiệm như: số lớp tích chập là 3, có 2 lớp max pooling và theo sau là 2 lớp kết hợp đầy đủ (Fully Connected). Số đầu ra của lớp softmax được điều chỉnh bằng số nhãn vận động bất thường trên từng tập dữ liệu. Để cải tiến hiệu suất huấn luyện và dự đoán, chúng tôi sử dụng kỹ thuật tối ưu Rectified Adam [19].

- *Mạng LSTM* [16]: được hiệu chỉnh để phù hợp cho các pha huấn luyện và dự đoán trên các tập dữ liệu thử nghiệm. Với đặc tính có thể nhớ thông tin trong một khoảng thời gian dài thì những đặc trưng ở mức cao trích chọn từ dữ liệu cảm biến được sử dụng hiệu quả tại bước dự đoán.

4. Kết quả và đánh giá

Chúng tôi sử dụng phương pháp kiểm chứng chéo 10 lần. Với phương pháp này, mỗi tập dữ liệu được chia thành 10 phần bằng nhau; 9 phần được lấy ra để huấn luyện và 1 phần được sử dụng để kiểm chứng. Quá trình này lặp lại cho đến khi cả 10 phần được kiểm chứng và kết quả được tính trung bình. Kết quả tổng thể được trình bày trong Bảng I. Trong Bảng I, SVM là bộ phân loại đã từng cho

Bảng I
KẾT QUẢ (F1-SCORE) TRÊN 4 TẬP DỮ LIỆU

PP/D.liệu	UTD	MobiFall	PTITAct	CMDFALL
SVM	0,85	0,79	0,87	0,45
CNN	0,94	0,88	0,91	0,82
LSTM	0,92	0,85	0,89	0,80
CNN-LSTM	**0,96**	**0,95**	**0,93**	**0,85**

kết quả khá tốt với các đặc trưng được trích chọn thủ công [9]. Tuy nhiên, so với các mô hình học sâu thì SVM thấp hơn đáng kể. Mô hình học sâu CNN với khả năng học các đặc trưng tự động tốt qua các phép tích chập giữa các bộ lọc, đã lựa chọn được các đặc trưng với đặc tính không gian (spatial) rất hiệu quả, đã cho kết quả tốt hơn đáng kể so với SVM. Mô hình LSTM cho kết quả tương đối tốt xấp xỉ với mô hình CNN. Mặc dù học và biểu diễn các đặc trưng không gian chưa phải là điểm mạnh của LSTM, nhưng với khả năng nhớ các thông tin theo chuỗi thời gian trong khoảng thời gian dài cũng giúp LSTM có khả năng dự đoán khá tốt, cạnh tranh được với CNN. Cuối cùng là mô hình đề xuất CNN-LSTM đã cho kết quả cao nhất 96% F1-score trên tập UTD, 95% trên tập MobiFall, 93% trên tập PTITAct, và 85% trên tập CMDFALL. Đây là kết quả cải tiến rất đáng kể so với 3 phương pháp còn lại. Điều này cũng cho thấy mô hình CNN-LSTM hiệu quả hơn hờ sự kết hợp của việc học và biểu diễn các đặc trưng của dữ liệu theo không-thời gian.

Bảng II
KẾT QUẢ CỦA MÔ HÌNH CNN-LSTM PHÁT HIỆN VẬN ĐỘNG BẤT THƯỜNG TRONG TẬP DỮ LIỆU CMDFALL

Tên hoạt động	**Precision**	**Recall**
ngã về phía sau	85,43	79,19
bò trên mặt đất	86,31	84,21
ngã về phía trước	89,56	87,58
ngã về bên trái	87,63	89,14
nằm trên giường và ngã về bên trái	70,42	67,30
nằm trên giường và ngã về bên phải	66,43	68,57
ngã về bên phải	91,62	92,25
ngồi trên ghế và ngã về bên trái	83,26	81,98
ngồi trên ghế và ngã về bên phải	79,12	78,67
nhảy loạng choạng	93,02	92,71
đi loạng choạng	84,25	82,59
Trung bình	**86,46%**	**83,59%**

Trong 4 tập dữ liệu kể trên thì tập UTD đơn giản nhất chỉ với 1 vận động bất thường (ngã), tiếp theo tập MobiFall với 4 vận động bất thường. Trong khi đó tập PTITAct và CMDFALL lần lượt là 8 và 11 vận động bất thường. Đặc biệt tập CMDFALL có nhiều vận động bất thường phức tạp hơn các tập dữ liệu khác nên điều này cũng lý giải kết quả các mô hình trên tập CMDFALL đều thấp hơn các tập dữ liệu khác. Bảng II trình bày kết quả chi tiết phát hiện vận động bất thường của mô hình đề xuất CNN-LSTM thử nghiệm trên tập CMDFALL. Kết quả ở Bảng II cho thấy, CNN-LSTM có thể đạt tới độ chính xác là 86,46% và độ bao phủ 83,59% trên tập dữ liệu CMDFALL. Đây cũng là kết quả tốt nhất so với các phương pháp khác. Một số vận động bất thường rất phức tạp như nằm trên giường và ngã cũng được phát hiện chính xác lên tới 70%. Trong khi đó các tư thế ngã về phía trước, ngã về bên phải, ngã về bên trái v.v...đều được phát hiện với độ chính xác xấp xỉ tới 90%.

V. KẾT LUẬN

Chúng tôi đã đề xuất một mô hình học sâu tích chập kết hợp với mạng bộ nhớ dài ngắn CNN-LSTM để giải quyết bài toán phát hiện các vận động bất thường của người sử dụng cảm biến đeo trên người. Kiến trúc đề xuất CNN-LSTM đã tận dụng được đặc tính không-thời gian của dữ liệu cảm biến để tự động học và biểu diễn các đặc trưng hiệu quả trên dữ liệu cảm biến không thuần nhất. Kết quả thử nghiệm trên 4 tập dữ liệu UTD, MobiFall, PTITAct và CMDFALL cho thấy mô hình đề xuất đã cho kết quả tốt hơn đáng kể so với các mô hình máy véc tơ hỗ trợ (SVM), mô hình học sâu tích chập (CNN) và mô hình mạng bộ nhớ dài ngắn (LSTM). Đặc việt với độ chính xác lên tới hơn 85% trên bộ dữ liệu CMDFALL cho thấy khả năng phát hiện tốt các vận động bất thường phức tạp. Kết quả này có

nhiều tiềm năng cho các ứng dụng hỗ trợ theo dõi người bệnh Parkinson, bệnh về vận động và người cao tuổi.

LỜI CẢM ƠN

Nghiên cứu này được hỗ trợ bởi Quỹ Phát triển Khoa học và Công nghệ Quốc gia (NAFOSTED) với mã số 102.04-2016.23.

TÀI LIỆU THAM KHẢO

[1] Hoey J, Plotz T, Jackson D, Monk A, Pham C, Olivier P (2011) *"Rapid specification and automated generation of prompting systems to assist people with dementia."* Pervasive and Mobile Computing 7(3):299-318, DOI 10.1016/j.pmcj.2010.11.007

[2] Gao Y, Long Y, Guan Y, Basu A, Baggaley J, Ploetz T (2019) *"Towards reliable, automated general movement assessment for perinatal stroke screening in infants using wearable accelerometers."* Proc ACM Interact Mob Wearable Ubiquitous Technol 3(1):12:1-12:22, DOI 10.1145/3314399

[3] Khan A, Mellor S, Berlin E, Thompson R, McNaney R, Olivier P, Plotz T (2015) *"Beyond activity recognition: Skill assessment from accelerometer data."* In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, UbiComp'15, pp 1155-1166, DOI 10.1145/2750858.2807534

[4] Pham C., Nguyen ST, Tran QH, Tran S, Vu H, Tran TH, Le TL (2020) *"SensCapNet: Deep neural network for non-obtrusive sensing based Human activity recognition."* IEEE Access 8:86934:86946, DOI 10.1109/ACCESS.2020.2991731

[5] Pham C, Diep NN, Phuong TM (2017) *"E-shoes: Smart shoes for unobtrusive human activity recognition."* In: 9th International Conference on Knowledge and Systems Engineering, KSE 2017, Hue, Vietnam, October 19-21, 2017, pp 269-274, DOI 10.1109/KSE.2017.8119470

[6] Pavllo D, Feichtenhofer C, Grangier D, Auli M (2019) *"3d human pose estimation in video with temporal convolutions and semi-supervised training."* In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[7] Sarita C, Mohd AK, Charul (2018) *"Multiple anomalous activity detection in videos."* In: Procedia Computer Science 125 (2018) pp. 336-345.

[8] Tran TH, Le T, Pham DT, Hoang VN, Khong VM, Tran QT, Nguyen TS, Pham C (2018) *"A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality."* pp 1947-1952, DOI 10.1109/ICPR.2018.8546308

[9] Nguyen, L., Le, A., T., Pham, C.; (2018) *"The Internet-of-Things based Fall Detection Using Fusion Feature."* In proc. of the 10th IEEE International Conference on Knowledge Systems Engineering (KSE). 129-134

[10] Ordonez F, Roggen D (2016) *"Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition."* Sensors 16(1):115

[11] Munzner S, Schmidt P, Reiss A, Hanselmann M, Stiefelhagen R, Durichen R (2017) *"Cnn-based sensor fusion tech-niques for multimodal human activity recognition."* In: Proceedings of the 2017 ACM International Symposium on Wearable Computers, pp 158-165

[12] Guan Y, Plotz T (2017) *"Ensembles of deep lstm learners for activity recognition using wearables."* Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1(2):1-28

[13] Ignatov A (2018) *"Real-time human activity recognition from accelerometer data using convolutional neural net-works."* Applied Soft Computing 62:915-922

[14] Dawar N, Kehtarnavaz N. (2018) *"A Convolutional Neural Network-Based Sensor Fusion System for Monitoring Transition Movements in Healthcare Applications."* In: proceeding of ICCA 482-485. 10.1109/ICCA.2018.8444326.

[15] Vavoulas G, Pediaditis M, Chatzaki C, Spanakis E, Tsiknakis Manolis, (2016) *"The MobiFall Dataset: Fall Detection and Classification with a Smartphone."* International Journal of Monitoring and Surveillance Technologies Research. 2. 44-56. 10.4018/ijmstr.2014010103.

[16] Liu J, Shahroudy A, Xu D, Wang G (2016) *"Spatio-temporal lstm with trust gates for 3d human action recognition."* In: European conference on computer vision, Springer, pp 816-833

[17] Chatzaki C, Pediaditis M, Vavoulas G, Tsiknakis M. (2017) *"Human Daily Activity and Fall Recognition Using a Smartphone's Acceleration Sensor."* 100-118. 10.1007/978-3-319-62704-5-7.

[18] Jinsung Y, Danial J, Mihaela VDS, (2019) *"Time-series Generative Adversarial Networks."* In: proc of 33rd conference on Neural Information Processing Systems (NeurIPS) pp.1-11.

[19] Liu L, et al. (2020) *"On the variance of the adaptive learning rate and beyond."* In proc. of the international conference on Learning Representation 2020. https://arxiv.org/pdf/1908.03265.pdf

[20] Hochreiter S, Schmidhuber J (1997) *"Long short-term memory."* Neural Computation 9(8):1735-1780, DOI 10.1162/neco.1997.9.8.1735

[21] Markham A, Trigoni N (2019) *"Selective sensor fusion for neural visual-inertial odometry."* In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 10542-10551

[22] Pham C, Nguyen TTT (2016) *"Real-time traffic activity detection using mobile devices."* In: Proceedings of the 10th ACM International Conference on Ubiquitous Information Management and Communication (IMCOM) 1-7

[23] Pham VC (2012) *"Human activity recognition for pervasive interaction."* PhD thesis. Newcastle University

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Tuấn Linh

Tốt nghiệp đại học ngành Công nghệ Thông tin, Đại học Giao thông Vận tải Hà Nội năm 2004. Nhận bằng Thạc Sỹ tại Đại học Thái Nguyên năm 2007.

Hiện là nghiên cứu sinh tại Học viện Công nghệ Bưu chính Viễn Thông.

Lĩnh vực nghiên cứu: kỹ thuật máy tính, điện toán tỏa khắp, các mô hình học máy và công nghệ cảm biến cho các ứng dụng chăm sóc sức khỏe.

Nguyễn Văn Thủy

Tốt nghiệp đại học năm 1999 ngành Điện tử - Viễn thông, trường Đại học Bách khoa Hà nội, nhận bằng Thạc sỹ năm 2005 tại ĐH bang New Mexico, Hoa Kỳ, Tiến sỹ năm 2012 tại ĐH Texas at Dallas, Hoa Kỳ.
Hiện công tác tại Học viện Công nghệ Bưu chính Viễn Thông.
Lĩnh vực nghiên cứu: học máy, lý thuyết thông tin, hệ thống thông tin thông minh, ứng dụng học sâu trong các hệ thống thông tin băng rộng.

Phạm Văn Cường

Tốt nghiệp đại học năm 1998 ngành Công nghệ Thông tin, Đại học Quốc gia Hà nội, nhận bằng Thạc sỹ năm 2005 tại ĐH bang New Mexico, Hoa Kỳ, Tiến sỹ năm 2012 tại ĐH Newcastle, Vương Quốc Anh.
Hiện là Phó giáo sư, giảng dạy tại khoa CNTT1, Học viện Công nghệ Bưu chính Viễn Thông.
Lĩnh vực nghiên cứu: học máy, điện toán tỏa khắp, tương tác người máy, nhận dạng hoạt động của người, các thuật toán học máy và công nghệ cảm biến cho các ứng dụng chăm sóc sức khỏe, thị giác máy tính, các công nghệ cảm biến, hệ thống nhúng và điều khiển.

Quan hệ giữa phụ thuộc hàm nới lỏng và phụ thuộc Boole dương tổng quát

Nguyễn Xuân Huy[1], Nguyễn Thị Vân[2], Trương Thị Thu Hà[3]
[1]Viện Hàn lâm Khoa học và Công nghệ Việt Nam
[2]Trường Cao đẳng Cộng đồng Hà Nội
[3]Trường Đại học Kinh doanh và Công nghệ
Tác giả liên hệ: Nguyễn Thị Vân, van.cdcd@gmail.com
Ngày nhận bài: 08/05/2020, ngày sửa chữa: 26/06/2020
Định danh DOI: 10.32913/mic-ict-research-vn.v2020.n1.926

Tóm tắt: Mục đích chính của bài báo là khảo sát và đặc tả các biến thể khác nhau của phụ thuộc Boole dương tổng quát. Tìm ra một đặc tả cho phụ thuộc nới lỏng tổng quát và chỉ ra rằng phụ thuộc nới lỏng nói chung và phụ thuộc hàm nới lỏng nói riêng chỉ là những trường hợp riêng của phụ thuộc Boole dương tổng quát.
Kết quả này được dùng làm cơ sở để xây dựng mô hình tổng quát về các lớp phụ thuộc khác nhau trong khai thác tri thức bằng các công cụ AI và học sâu.

Từ khóa: *phụ thuộc Boole dương tổng quát, phụ thuộc hàm nới lỏng.*

Title: **Relationship between Relaxed Functional Dependencies and Generalized Positive Boolean Dependencies**

Abstract: The main purpose in this paper is studying and descripting the variety of the generalized positive Boolean dependencies. A class of genneralized relaxed dependencies is introduced and pointed out that this class and class of relaxed functional dependencies are partial of the generalized positive Boolean dependencies.
The result can be used to construct a model of different dependencies for knowledge mining with the AI approach and deep learning.

Keywords: *Generalized Boolean Dependencies, Relaxed Functional Dependencies.*

I. ĐẶT VẤN ĐỀ

Phụ thuộc dữ liệu đóng vai trò quan trọng trong thiết kế các hệ cơ sở dữ liệu và khai thác tri thức. Việc phát hiện các loại phụ thuộc khác nhau trong dữ liệu là một vấn đề nghiên cứu thú vị, có ý nghĩa và cũng là một trong những mục tiêu của lĩnh vực đảm bảo ngữ nghĩa, tính nhất quán và khai thác tri thức trên văn bản.

Phụ thuộc dữ liệu hay ràng buộc dữ liệu là những mệnh đề qui định các tính chất của dữ liệu cần được tôn trọng trong quá trình xử lý và lưu trữ nhằm phản ánh đúng hiện thực khách quan. Sau công bố năm 1970 của Codd về phụ thuộc hàm trong cơ sở dữ liệu, các nhóm nghiên cứu đã phát triển và đề xuất nhiều loại phụ thuộc dữ liệu ngày càng tinh tế hơn nhằm mở rộng và nâng cao khả năng mô tả dữ liệu trong thế giới thực. Một trong những hướng nghiên cứu quan trọng là những phụ thuộc hàm nới lỏng $X \rightarrow Y$, trong đó X và Y là các các tập thuộc tính, $\rightarrow$ thể hiện sự phụ thuộc của tập thuộc tính Y vào tập thuộc tính X với các điều kiện nới lỏng khác nhau [5].

Ví dụ, trong giao thông ta thường có các phụ thuộc dữ liệu sau:

Bảng I
BẢNG GIÁ TAXI

Xe	Điểm đi	Điểm đến	Khoảng cách	Thành tiền
201	18 Hoàng Quốc Việt	20 Nguyễn Trãi	7,3 km	63.000
67	20 Hoàng Văn Thái	125 Nguyễn Phong Sắc	5,5 km	50.000
120	18 Hoàng Quốc Việt	18 Nguyễn Trãi	7 km	63.000
45	42 Vương Thừa Vũ	78 Trần Đăng Ninh	5,3 km	50.000

Theo phụ thuộc hàm kinh điển: *Khoảng cách* $\rightarrow$ *Thành tiền* thể hiện ngữ nghĩa sau: Nếu hai chuyến taxi chở khách đi cùng một khoảng cách thì hai xe nhận được cùng một số tiền.

Phụ thuộc hàm nới lỏng: *Khoảng cách (0,2)* $\rightarrow$ *Thành*

tiền (0) thể hiện ngữ nghĩa sau: Nếu hai chuyến taxi chở khách có khoảng cách chênh lệch không quá 0,3 km thì hai xe nhận được số tiền như nhau.

Năm 2016, nhóm tác giả Loredana Caruccio, Vincenzo Deufemia, Giuseppe Polese đã tập hợp và phân loại các phụ thuộc nới lỏng (relaxed functional dependencies) và phân tích các đặc trưng của một số dạng nới lỏng [5].

Năm 1992, nhóm tác giả Nguyễn Xuân Huy và Lê Thị Thanh đã đề xuất một loại phụ thuộc dữ liệu mới là phụ thuộc Boole dương tổng quát (PTBDTQ) và chứng minh phụ thuộc hàm và một số biến thể của phụ thuộc hàm như phụ thuộc yếu, phụ thuộc mạnh, phụ thuộc đối ngẫu là những trường hợp riêng của phụ thuộc Boole dương tổng quát [1, 3]. Tiếp đến trong các năm từ 2012 đến 2016 nhóm nghiên cứu về phụ thuộc Boole dương tổng quát đã tiếp tục chứng minh rằng các phụ thuộc sai khác [6] và các biến thể như các phụ thuộc hàm xấp xỉ [4] cũng là những trường hợp riêng của phụ thuộc Boole dương tổng quát.

Trong bài này, tiếp tục mạch nghiên cứu trên, sẽ chỉ ra rằng các phụ thuộc hàm nới lỏng cũng là những trường hợp riêng của phụ thuộc Boole dương tổng quát.

II. CÁC KHÁI NIỆM VÀ QUY ƯỚC

Cho U = { x_1, ..., x_n } là tập hữu hạn các biến Boole nhận giá trị trong tập trị logic $\mathbb{B} = \{0, 1\}$ Tập các công thức Boole (CTB), kí hiệu L(U), bao gồm các biểu thức được xây dựng từ các biến trong U, các hằng 0/1 và các phép toán logic $\wedge, \vee, \neg, \rightarrow$

Mỗi vector 0/1, $v = (v_1,...,v_n)$ trong không gian $\mathbb{B}^n$ được gọi là một phép gán trị. Khi đó với mỗi CTB $f = L(U)$, $f(v)$ là trị của công thức f đối với phép gán trị v. Kí hiệu e là phép gán trị đơn vị, $e = (1,1,...,1)$. Công thức $f = L(U)$ được gọi là công thức Boole dương (CTBD) nếu $f(e)= 1$.

Ký hiệu $P(U)$. Với mỗi công thức Boole $f \in L(U)$, kí hiệu $T_f = \{ v \in \mathbb{B}^n | f(v) = 1 \}$ là bảng *chân lí của f*. Mỗi tập công thức $F \subseteq L(U)$ được hiểu là một hội logic của các công thức thành phần, $F = \wedge f \mid f \in F$. Khi đó, $T_F = \bigcap T_f \mid f \in F$ là *bảng chân lí của tập công thức F*. Với hai CTBD f và g trên U, ta đã biết CTBD g được suy dẫn logic từ CTBD f và được ký hiệu là $f \rightarrow g$ khi và chỉ khi $T_f \in T_g$. Tương tự, nếu F là tập các CTBD trên U thì $F \rightarrow g$ khi và chỉ khi $T_F \subseteq T_g$. Hai tập CTBD F và G trên U là tương đương nhau khi và chỉ khi $T_F = T_G$.

Theo truyền thống của lý thuyết cơ sở dữ liệu, ta chấp nhận các ký hiệu sau đây:

Hợp của hai tập X và Y được viết XY; giao của hai tập X và Y được viết $X \bigcap Y$; phép trừ hai tập X và Y được ký hiệu là X-Y.

Phép hội hai công thức logic X và Y được kí hiệu là $X \wedge Y$ hoặc XY hoặc X và Y (trong trường hợp cần chỉ rõ dấu phép toán) và được gọi là tích (logic); phép tuyển hai công thức logic X và Y được ký hiệu $X \vee Y$ hoặc $X+Y$ và được gọi là tổng (logic); dấu ' thay cho phép phủ định ¬. Ví dụ, biểu thức $ab + c + bd$ ' là một biểu diễn tương đương với công thức logic truyền thống a ∧ b ∨ c ∨ b ∧(¬d).

Ví dụ, cho tập U={a,b,c} khi đó công thức f:$ab \rightarrow c$ là một công thức Boole dương vì $f(1,1,1) = 1$.

Bảng II
BẢNG CHÂN LÝ T_f

a	b	c	ab	f=$ab \rightarrow c$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Nếu U = { x_1, x_2, …, x_n } là tập các thuộc tính và v là một bộ trong quan hệ r trên U, $x \in U$ thì $v.x$ được ký hiệu là trị của thuộc tính x trong bộ v. Nếu $X \subseteq U$ thì ta định nghĩa $v.X = \{v.x \mid x \in X\}$. Với tập thuộc tính U cho trước, quan hệ trên tập thuộc tính U được kí hiệu là r, khi cần chỉ rõ tập thuộc tính ta có thể sử dụng kí hiệu $r(U)$.

Các qui ước và kí pháp truyền thống khác được giới thiệu trong các tài liệu kinh điển về cơ sở dữ liệu và được trích dẫn trong [1] và [2].

III. PHỤ THUỘC BOOLE DƯƠNG TỔNG QUÁT

Cho U = {x_1, ..., x_N } là tập hữu hạn các biến Boole nhận giá trị trong tập trị logic $\mathbb{B} = \{0, 1\}$.

Ta quy ước mỗi miền trị d_x của thuộc tính x trong U có chứa ít nhất hai phần tử. Với mỗi miền trị d_x, xét ánh xạ $\propto_x$: $d_x^2 \rightarrow \mathbb{B}$ thoả các tiên đề sau [2][3]:

$\forall a, b \in d_x$

A1) Tiên đề phản xạ $\propto_x(a, a) = 1$

A2) Tiên đề đối xứng $\propto_x(a, b) = \propto_x(b, a)$

A3) Tiên đề bộ phận $\exists c \in d_x$: $\propto_x(a, c) = 0$

$\propto_x$ chính là quan hệ (hai ngôi) bộ phận thực sự, thoả các tính chất phản xạ và đối xứng trên miền trị d_x. Việc xác định $\propto_x$ được hiểu là thiết lập một *phép sánh trị* trên miền trị d_x cho thuộc tính x

Quan hệ bằng $=_x$ (tạm gọi là quan hệ đẳng thức) được định nghĩa: $\forall a, b \in d_x$: $=_x(a, b) = 1$, khi và chỉ khi $a = b$, là trường hợp riêng của phép sánh trị và được ngầm định trong rường hợp không định nghĩa tường minh phép sánh trị cho thuộc tính x.

Ta gọi lược đồ dữ liệu là một cặp $p = (U, F)$, trong đó U là tập thuộc tính với các miền trị tương ứng và các phép sánh trị trên mỗi miền trị, F là tập các phụ thuộc trên U [1, 2, 3].

Cho lược đồ $p = (U, F)$ và quan hệ r trên U. Với mỗi cặp bộ $u = (u_1, u_2,... , u_n)$, $v = (v_1, v_2,... , v_n)$ trong r, ta đặt tương ứng một vector 0/1 $t = (t_1, t_2,... , t_n) \in \mathbb{B}_n$ và kí hiệu là $t = \propto(u, v)$, trong đó thành phần $t.x$ ứng với thuộc tính x trong U chính là ảnh của ánh xạ $t.x = \propto_x(u.x, v.x)$.

Khi đó mỗi quan hệ r sẽ được đặt tương ứng với tập các vector 0/1, $T_r = \{ \propto(u, v) \mid u, v \in r\}$, và được gọi là *bảng trị* của quan hệ r trên LĐBDTQ p [2, 3].

Quan hệ r trên tập thuộc tính U thỏa PTBDTQ f (tập PTBDTQ F) và viết $r(f)$ $(r(F))$ nếu $T_r \subseteq T_f$ $(T_r \subseteq T_F)$.

Mỗi công thức Boole dương f trong $P(U)$ với các phép sánh trị cho trước được gọi là một phụ thuộc Boole dương tổng quát (PTBDTQ), lược đồ thu được trong trường hợp này được gọi là lược đồ với phụ thuộc boole dương tổng quát.

IV. PHỤ THUỘC HÀM NỚI LỎNG

Phụ thuộc hàm nới lỏng [5] được xây dựng dưới dạng thức chung: f: $X(\lambda) \rightarrow Y(\gamma)$; $X, Y \subseteq U$ với các điều kiện nới lỏng λ và γ như sau:

Quan hệ $r(U)$ thỏa PTHNL f: $X(\lambda) \rightarrow Y(\gamma)$; $X, Y \subseteq U$ nếu $T_r \subseteq T_f$

Nới lỏng các phép sánh trị trên một vài thuộc tính: ngoài sánh trị đẳng thức có thể xét các phép xấp xỉ theo độ đo, hoặc các phép so sánh $<, \leq, >, \geq, =, \neq$. Quan hệ r thỏa phụ thuộc hàm nới lỏng theo phép sánh trị $X\ (\rho)\rightarrow Y$ và được viết là $r(X\ (\rho)\rightarrow Y)$ khi và chỉ khi hai bộ bất kỳ trong r sai khác nhau không quá ngưỡng ρ trên X thì hai bộ đó cũng sai khác nhau không quá ngưỡng ρ trên Y.

Nới lỏng theo lực lượng:

$$r(X\ (\delta)\rightarrow Y) = \frac{\neq r(X \rightarrow Y)}{\neq r} \leq \delta$$

Trong đó $\neq S$ là số phần tử có trong tập S. Biểu thức trên cho biết quan hệ r thỏa phụ thuộc hàm nới lỏng theo lực lượng $X\ (\delta)\rightarrow Y$ và được viết là $r(X\ (\delta)\rightarrow Y)$ khi và chỉ khi tỷ lệ giữa số lượng các bộ còn lại thỏa phụ thuộc hàm truyền thống $X \rightarrow Y$ và các bộ của quan hệ r đạt trên ngưỡng δ. Nói cách khác, ta có thể xóa khỏi quan hệ r một số bộ với tỷ lệ tỷ lệ δ để quan hệ còn lại thỏa phụ thuộc hàm truyền thống $X \rightarrow Y$.

Định nghĩa chung về *phụ thuộc hàm nới lỏng* tổng quát như sau:

Cho U là tập thuộc tính, X và Y là hai tập thuộc tính trong U. PTH nới lỏng có dạng:

$$f(X\ (\gamma)\rightarrow Y(\vartheta),\ \ X, Y \subseteq U$$

Ta nói phụ thuộc hàm nới lỏng f thỏa trong quan hệ $r(U)$ nếu:

$$f(X\ (\gamma)\rightarrow Y(\vartheta),\ \ X, Y \subseteq U$$

Với mọi cặp bộ $u, v \in r$, nếu tân từ $\gamma(u.X,v.X)$ suy ra được tân từ $\vartheta(u.X,v.Y)$.

Phụ thuộc hàm nới lỏng được hiểu là phụ thuộc hàm với điều kiện kèm theo nhằm giảm nhẹ các điều kiện của phụ thuộc hàm chính thống. Các điều kiện giảm nhẹ được phát biểu thông qua các tân từ γ và ϑ

V. CÁC LỚP BIẾN THỂ CỦA PHỤ THUỘC BOOLE DƯƠNG TỔNG QUÁT

Các kết quả chủ yếu của mục này cho thấy, tùy thuộc vào các dạng của công thức Boole dương cụ thể và phép sánh trị alpha cụ thể ta có thể nhận được các phụ thuộc nới lỏng khác nhau ứng với các lược đồ dữ liệu tương ứng sau đây.

1. Lớp IE *(Implication Fomula và Equal Comparison)*

Lớp IE là lớp các lược đồ phụ thuộc hàm kinh điển, được xây dựng trên cơ sở phép toán suy dẫn và phép sánh trị đẳng thức.

Cho tập các thuộc tính $U = (x_1, x_2,... , x_n)$, $n \geqslant 1$. Giả sử X, $Y \subseteq U$. Một phụ thuộc thuộc lớp IE là biểu thức dạng f: X$\rightarrow Y$.

Dựa trên biểu thức logic X và Y ta phân biệt các dạng phụ thuộc hàm sau:

IE-1: Biểu thức logic có dạng $X \rightarrow Y$, ta có phụ thuộc hàm truyền thống: Cho LĐQH (U, F) Cho LĐQH (U, F) và một phụ thuộc hàm f:$X\rightarrow Y$ trên U. Ta nói quan hệ r thỏa phụ thuộc hàm f và ký hiệu $r(f)$, nếu hai bộ tùy ý trong r giống nhau trên X thì cũng giống nhau trên Y

IE-2: biểu thức logic có dạng $\bigvee X \rightarrow Y$ cho ta lược đồ phụ thuộc hàm mạnh: Cho quan hệ r trên tập thuộc tính U, quan hệ R thỏa phụ thuộc hàm mạnh $\bigvee X \rightarrow Y$ trên U nếu với hai bộ u, v bất kỳ trong r: u và v bằng nhau tại một thuộc tính nào đó trên X thì u và v cũng bằng nhau trên Y.

IE-3: Biểu thức logic có dạng $X \rightarrow \bigvee Y$, ta có lược đồ phụ thuộc hàm yếu: Cho quan hệ r trên tập thuộc tính U, quan hệ r thỏa phụ thuộc hàm yếu $X \rightarrow \bigvee Y$ trên U nếu với hai bộ u, v bất kỳ trong r: u và v bằng nhau trên X thì u và v cũng bằng nhau tại một thuộc tính nào đó trên Y.

IE-4: Biểu thức logic có dạng $\bigvee X \rightarrow \bigvee Y$, ta có lược đồ phụ thuộc hàm đối ngẫu: Cho quan hệ r trên tập thuộc tính U, quan hệ r thỏa phụ thuộc hàm đối ngẫu $\bigvee X \rightarrow \bigvee Y$ trên U nếu với hai bộ u, v bất kỳ trong r: u và v bằng nhau

tại một thuộc tính nào đó trên X thì u và v cũng bằng nhau tại một thuộc tính nào đó trên Y.

Bảng dưới đây tóm lược các đặc tả cho các lớp con IE1-4 của lớp IE

Dạng phụ thuộc	**Tên gọi**	**Đặc điểm**
$X \rightarrow Y$	phụ thuộc hàm	ràng buộc chặt
$\bigvee X \rightarrow Y$	phụ thuộc hàm mạnh	nới lỏng về trái
$X \rightarrow \bigvee Y$	phụ thuộc hàm yếu	nới lỏng về phải
$\bigvee X \rightarrow \bigvee Y$	phụ thuộc hàm đối ngẫu	nới lỏng
$X(\delta) \rightarrow Y$	phụ thuộc xấp xỉ	nới lỏng

Như vậy, các dạng phụ thuộc hàm IE-1, IE-2, IE-3, IE-4 là các phụ thuộc hàm nới lỏng và chúng là những trường hợp riêng của phụ thuộc Boole dương tổng quát.

2. Lớp LA *(Logic Fomula và Alpha Comparison)*

Dựa theo điều kiện nới lỏng, ta chia lớp LA thành các lớp con sau đây:

LA-1: Lớp LA-1 là lớp các phụ thuộc được xây dựng trên cơ sở phép toán suy dẫn và phép sánh trị alpha.

Cho tập các thuộc tính $U = (x_1, x_2, \ldots, x_n)$, $n \geq 1$. Giả sử $X, Y \subseteq U$. Một phụ thuộc thuộc lớp LA-1 là biểu thức dạng f: $\alpha_X \rightarrow \alpha_Y$ với các phép sách trị α như đã trình bày ở phần III.

Ta nói quan hệ r thỏa LA-1: $\alpha_X \rightarrow \alpha_Y$ và viết $r(X(\alpha) \rightarrow Y(\alpha))$, nếu với hai bộ bất kỳ $u, v \in r$, thỏa các ràng buộc được đặc tả bởi α_X trên tập thuộc tính X, thì u và v cũng thỏa các ràng buộc được đặc tả bởi hàm sai khác α_Y trên tập thuộc tính Y:

$$r(X(\alpha) \rightarrow Y(\alpha)) \overset{def}{\Longleftrightarrow} \forall u,v \in r\colon \alpha(u.X, v.X) \Rightarrow \alpha(u.Y, v.Y)$$

Trong lớp LA-1, ta có các phụ thuộc đại diện sau:

- Nếu biểu thức logic dạng f: $\alpha_X \rightarrow \alpha_Y$, ta có phụ thuộc sai khác [6], với α_X và α_Y là các hàm sai khác được định nghĩa :

Cho quan hệ r trên tập thuộc tính U, $a \in U$ và độ sai khác m_a. Hàm sai khác α_a trên thuộc tính $\propto$ đặc tả ràng buộc của độ sai khác m_a: Với hai trị $x, y \in d_a$, ta định nghĩa $\alpha_a(x, y) = 1$ khi và chỉ khi $m_a(x, y)$ thỏa điều kiện cho α_a dưới dạng các biểu thức so sánh với các phép so sánh $=, \neq, <, \leqslant, >$ và $\geqslant$.

Cho tập thuộc tính $X \subseteq U$. Hàm sai khác $\emptyset_X$ trên tập thuộc tính X là hội logic của các hàm sai khác trên mọi thuộc tính $a \in X$: $\emptyset_X = \bigwedge_{\alpha \in X} \emptyset_\alpha$

Trong [5], đã chỉ ra rằng phụ thuộc hàm sai khác là trường hợp riêng của phụ thuộc hàm nới lỏng, trong phần này chỉ ra rằng, phụ thuộc sai khác thuộc lớp LA-1.

- Nếu biểu thức logic có dạng f: $X(\delta) \rightarrow Y$, $0 \leqslant \delta \leqslant 1$ ta có *phụ thuộc nới lỏng theo lực lượng*.

Ví dụ, khi khảo sát các khoa của một trường đại học người ta đánh giá tổng thể môi trường đào tạo của khoa theo ý kiến của các chuyên gia (thuộc tính A). Có 4 mức cho A lần lượt là a, b, c và d, trong đó a là mức tốt nhất. Thuộc tính B cho biết mức độ hài lòng của các sinh viên về trường đó. B gồm 4 mức là 1, 2, 3 và 4, trong đó 1 là mức cao nhất. Giả sử kết quả khảo sát được thể hiện trong quan hệ $r(A, B)$. Người ta muốn biết giữa môi trường học tập (A) và mức độ hài lòng (B) có đạt trên ngưỡng 0.6 hay không, tức là xác định f: $A(0,6) \rightarrow B$?

r

A	B	
a	1	
b	2	
a	1	
a	2	*
b	2	

Ta thấy, sau khi xóa bộ thứ tư khỏi quan hệ r thì quan hệ con còn lại, ký hiệu là r' sẽ thỏa phụ thuộc hàm truyền thống $A \rightarrow B$. Tỷ lệ thu được lúc này sẽ là

$$\frac{\neq r'}{\neq r} = \frac{4}{5} = 0,8 > 0,6$$

- Phép sánh trị theo khoảng

Miền trị d_A được phân hoạch thành k khoảng không giao nhau:

$$d_A = \bigcup_{i=1}^{k} [a_i\,; b_i]$$

$$[a_i; b_i] \cap [a_j; b_j] = \emptyset, 1 \leq i, j \leq k, i \neq j$$

$\forall$ x, y $\in d_A$: α_A)(x, y) = 1 khi và chỉ khi x và y thuộc cùng một khoảng:

Ví dụ, điểm tổng kết của học sinh được chia thành 8 khoảng

Điểm F : [0; 3,9]
Điểm D : [4,0; 4,7]
Điểm D+ : [4,8; 5,4]
Điểm C : [5,5; 6,2]
Điểm C+ : [6,3; 6,9]
Điểm B : [7,0; 7,7]

Điểm B+ : [7,8; 8,4]
Điểm A : [8,5; 10]

Lớp LA-2: Lớp LA-2 là lớp các phụ thuộc được xây dựng trên cơ sở phép toán logic Boole và phép sánh trị đẳng thức. Đại diện cho lớp LA-2 là phụ thuộc Boole dương. [2, 3]

Lớp LA-3: Lớp LA-3 là lớp các phụ thuộc được xây dựng trên cơ sở phép toán logic Boole và phép sánh trị alpha (mục 4.2). Đại diện cho lớp LA-3 là phụ thuộc Boole dương tổng quát (phần III), đây cũng là lớp phụ thuộc bao hàm các phụ thuộc logic trong cơ sở dữ liệu đã và đang được nghiên cứu bởi các nhóm tác giả trong và ngoài nước.

VI. KẾT LUẬN

Việc phân loại và đề xuất một mô hình chung cho các loại phụ thuộc dữ liệu là một trong những vấn đề đang được giới nghiên cứu dữ liệu lớn quan tâm. Các điều kiện cần và đủ để nhận biết các đặc trưng của một lớp phụ thuộc là cơ sở để phân loại và tạo ra mối liên hệ giữa các lớp phụ thuộc.

Bài báo mô tả mối quan hệ giữa phụ thuộc hàm nới lỏng và phụ thuộc Boole dương tổng quát trong cơ sở dữ liệu. Các kết quả chủ yếu bao gồm:

1. Xây dựng các lớp phụ thuộc trong cơ sở dữ liệu dựa vào hai đặc trưng cơ bản là công thức suy dẫn và phép sánh trị.

2. Chỉ ra mối quan hệ giữa phụ thuộc hàm nới lỏng với phụ thuộc Boole dương tổng quát. Từ các loại phụ thuộc hàm nới lỏng có thể tổng quát hóa thành phụ thuộc Boole dương tổng quát và ngược lại, từ lý thuyết phụ thuộc Boole dương tổng quát có thể thực tế hóa về các phụ thuộc hàm nới lỏng tùy thuộc từng trường hợp biến thể của yêu cầu thực tế.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Xuân Huy (2006), *"Các phụ thuộc logic trong cơ sở dữ liệu"*, NXB Thống Kê.

[2] Berman J., Blok W. J. (1988), *"Positive Boolean dependencies"* Inf. Processing Letters, 27, p.147 – 150.

[3] Huy Nguyen Xuan, Thanh Le Thi (1992), *"Generalized Positive Boolan Dependencies"*, J. Inf. Process. Cybern. EIK, vol. 28, p. 363 – 370.

[4] Jalal Atoum (2009), *"Mining Approximate Functional Dependencies from Databases"*, European Journal of Scientific Research, ISSN 1450 – 216X vol. 33 no. 2, p.338 – 346.

[5] Loredana Caruccio, Vincenzo Deufemia, Giuseppe Polese (2016), *"Relaxed Functional Dependencies - A Survey of Approaches"*, IEEE Transactions on Knowledge and Data Engineering, vol. 28, p. 147–165.

[6] Song S. and Chen L. (2011), *"Differential Dependencies: Reasoning and Discovery"*, *ACM Trans. Datab. Syst*, vol.9, no 4, Article 39.

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Xuân Huy

Sinh năm 1944 tại Hải Phòng. Cử nhân Toán, Đại học Sư phạm Leningrad (Liên Xô) năm 1973. Tiến sỹ CNTT năm 1982, tiến sỹ khoa học CNTT năm 1990, Viện Hàn lâm Khoa học Liên Xô.

Nguyên trưởng Phòng Cơ sở dữ liệu và Lập trình, Viện Công nghệ Thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam (1997-2009). Lĩnh vực nghiên cứu: Cơ sở dữ liệu và Công nghệ phần mềm. Email: nxhuy564@gmail.com.

Nguyễn Thị Vân

Sinh năm 1985 tại Hà Tĩnh. Cử nhân CNTT tại Trường Đại học Kinh doanh và Công nghệ Hà Nội năm 2011. Thạc sĩ ngành Khoa học máy tính tại Trường Đại học Công nghệ thông tin và Truyền thông năm 2014. Hiện đang công tác tại Khoa Công nghệ thông tin, Trường Cao đẳng Cộng đồng Hà Nội. Lĩnh vực nghiên cứu: Các phụ thuộc logic trong cơ sở dữ liệu, mô hình dữ liệu và cơ sở dữ liệu. Email: van.cdcd@gmail.com

Trương Thị Thu Hà

Sinh năm 1979 tại Nghệ An. Cử nhân CNTT Trường Đại học Sư phạm Hà Nội năm 2000. Thạc sĩ CNTT Trường Đại học Công nghệ, Đại học Quốc Gia Hà Nội năm 2006. Tiến sĩ CNTT, Học viện Kỹ thuật Quân sự năm 2018. Hiện công tác tại Trường Đại học Kinh doanh và Công nghệ Hà Nội. Lĩnh vực nghiên cứu: Cơ sở dữ liệu, Công nghệ phần mềm. Email: thuha.bh@gmail.com

Research, Development and Application on
Information and Communication Technology
http://www.ictmag.vn
Tap 2020, So 1
June

TABLE OF CONTENTS