Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Nha Trang, ngày 8-9/10/2020
DOI: 10.15625/vap.2020.00189

NGĂN CHẶN THÔNG TIN SAI LỆCH NHIỀU CHỦ ĐỀ TRÊN MẠNG XÃ HỘI TRỰC TUYẾN

Phạm Văn Dũng[1], Nguyễn Thị Tuyết Trinh[2], Nguyễn Việt Anh[1]
[1]Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam
[2]Học viện Y Dược học cổ truyền Việt Nam
pvdungc500@gmail.com, trinhnt83@gmail.com, anhnv@iot.ac.vn

TÓM TẮT: *Những năm gần đây, mạng xã hội trực tuyến (MXH) đang trở thành phương tiện truyền thông phổ biến trên thế giới. Bên cạnh những lợi ích mà MXH mang lại thì nó cũng cho phép lan truyền nhanh chóng thông tin sai lệch thuộc nhiều chủ đề khác nhau như: Văn hóa, chính trị, kinh tế,... làm ảnh hưởng tiêu cực đến người dùng. Để hạn chế ngăn chặn ảnh hưởng của thông tin sai lệch trên MXH. Trong bài báo này, chúng tôi đề xuất giải pháp tìm ra một tập nút mà khi loại bỏ chúng khỏi mạng thì hạn chế tối đa ảnh hưởng của thông tin sai lệch trên MXH với thời gian và chi phí cho phép. Bài toán được chứng minh là NP-khó ngay cả trong trường hợp MXH là cây có gốc tại nút phát tán thông tin sai lệch duy nhất và tính toán hàm mục tiêu là #P-khó ngay cả khi tập cần xóa chỉ có một nút. Chúng tôi cũng chứng minh hàm mục tiêu có tính chất monotone và submodular, dựa vào tính chất này chúng tôi đề xuất thuật toán Tham lam GA (Greedy Algorithm) cho tỷ lệ xấp xỉ* $(1 - 1/\sqrt{e})$. *Chúng tôi tiếp tục đề xuất thuật toán Tham lam tăng tốc FGA (Fast Greedy Algorithm) dựa trên đồ thị không có chu trình và vai trò ảnh hưởng của các nút, chúng tôi cho thấy tính toán hàm mục tiêu được thực hiện trong thời gian tuyến tính và thuật toán FGA có thể áp dụng đối với mạng lên đến đến hàng triệu cạnh. Các thực nghiệm được tiến hành trên dữ liệu của mạng xã hội thực, kết quả cho thấy hiệu suất của các thuật toán chúng tôi đề xuất vượt trội hơn các thuật toán cơ sở khác.*

Từ khóa: *Mạng xã hội, tối ưu hóa, lan truyền thông tin, ngăn chặn thông tin, thông tin sai lệch.*

I. GIỚI THIỆU

Những năm gần đây, **MXH** đã trở thành một nền tảng mạnh mẽ trong truyền thông số, đóng góp đáng kể vào sự phát triển của thế giới. Tuy nhiên, sự lan truyền thông tin sai lệch trên **MXH** ảnh hưởng không nhỏ đến kinh tế, chính trị, xã hội và tác động tiêu cực đến cộng đồng [1, 2]. Do đó, cần có những giải pháp tối ưu để hạn chế ảnh hưởng của thông tin sai lệch trên **MXH**, gọi chung là ngăn chặn ảnh hưởng **IB** (Influences blocking - IB). Hai phương pháp phổ biến để giải quyết bài toán **IB** là: 1) Xóa bỏ một tập nút (cạnh) hoặc tiêm vắc xin (theo ngôn ngữ dịch tế học) vào tập nút (cạnh) để hạn chế ảnh hưởng của thông tin sai lệch. Ví dụ, các tác giả trong [3, 4] đề xuất phương pháp Heuristic để loại bỏ các cạnh nhằm giảm thiểu ảnh hưởng của một nhóm thông tin sai lệch. Dưới góc độ dịch tế học, một số nghiên cứu khác đưa ra chiến lược tiêm vắc xin vào nhóm các nút (cạnh) để giảm sự lây lan của dịch bệnh [5-7],.... 2) Tẩy nhiễm thông tin, bằng cách chọn tập nút để phát tán thông tin tích cực nhằm chống lại thông tin tiêu cực. Theo hướng này, các tác giả trong [8-10] nghiên cứu chọn k nút để lan truyền thông tin tích cực để tẩy nhiễm thông tin xấu,... Các tác giả trên đều xem xét mọi thông tin sai lệch là như nhau. Trên thực tế thì thông tin sai lệch rất đa dạng, thuộc nhiều chủ đề và ảnh hưởng của những thông tin này đến mỗi người dùng là khác nhau. Ngoài ra, các tác giả xem xét chi phí loại bỏ các nút là như nhau, nhưng thực tế thì mỗi nút có vai trò, ảnh hưởng khác nhau trên mạng, nên chi phí xóa bỏ các nút cũng khác nhau. Trong nghiên cứu [11,12], các tác giả cũng đã đề cập đến tác động của các chủ đề trong việc lan truyền thông tin. Thông tin nhiều chủ đề hay nhiều khía cạnh cũng đã được xem xét trong các nghiên cứu bài toán tối đa hóa ảnh hưởng **IM** (Infuence Maximum - IM) [13-15]. Gần đây trong [16], các tác giả đã nghiên cứu ngăn chặn thông tin sai lệch nhiều chủ đề có ràng buộc về chi phí. Vì vậy, giải quyết bài toán **IB** theo hướng xem xét tác động của từng chủ đề thông tin sai lệch đến người dùng là một điều hết sức cần thiết, mang tính thực tiễn cao.

Trong bài báo này chúng tôi đề xuất bài toán: **Ngăn chặn thông tin sai lệch nhiều chủ đề trên mạng xã hội trực tuyến MMBO** (Multi-topic Misinformation Blocking on Online social networks - MMBO), bài toán có xét đến ràng buộc về thời gian và chi phí. Những đóng góp của bài báo như sau:

a. Để mô tả quá trình lan truyền thông tin nhiều chủ đề trên **MXH**, chúng tôi đề xuất mô hình Ngưỡng tuyến tính nhiều chủ đề có ràng buộc thời gian **MLT** (**M**ulti-topics **L**inear threshold model has **T**ime constraints - MLT) bằng cách mở rộng mô hình Ngưỡng tuyến tính **LT** (Linear Threshold model) [17]. Trên mô hình này, thông tin được lan truyền đến các nút dựa trên tổng ảnh hưởng của các nút hàng xóm đến các nút đó và ngưỡng kích hoạt theo các chủ đề của thông tin khác nhau.

b. Chúng tôi chỉ ra rằng **MMBO** thuộc lớp bài toán NP- khó, tính toán độ giảm ảnh hưởng khi xóa bỏ tập nút (hàm mục tiêu) là #P - khó và hàm mục tiêu có tính chất monotone và submodular. Dựa trên tính chất này chúng tôi đề xuất thuật toán Tham lam **GA** cho tỷ lệ xấp xỉ $(1 - 1/\sqrt{e})$.

c. Chúng tôi tiếp tục đề xuất phương pháp giải quyết bài toán **MMBO** dựa trên đồ thị không có chu trình (Directed Acyclic Graph - DAG) và cho thấy tính toán hàm mục tiêu trên **DAG** có thời gian đa thức. Chúng tôi đề xuất thuật toán Tham lam tăng tốc **FGA** dựa trên vai trò ảnh hưởng của các nút và cập nhật nhanh hàm mục tiêu trên cấu trúc **DAG**, thuật toán có thể áp dụng đối với đồ thị lên đến hàng triệu cạnh.

Các thử nghiệm được thực hiện trên các mạng xã hội thực bao gồm NetHepP, Epinions và Amazon cho thấy các thuật toán được đề xuất vượt trội hơn các thuật toán cơ sở khác về cả về hiệu xuất và thời gian thực hiện. Bài báo được

trình bày gồm 06 phần, ngoài phần tóm tắt và tài liệu tham khảo, bài báo bao gồm Phần I - Giới thiệu, Phần II - Mô hình và định nghĩa bài toán, Phần III - Đề xuất thuật toán, Phần IV - Kết quả thực nghiệm và Phần V - Kết luận.

II. MÔ HÌNH VÀ ĐỊNH NGHĨA BÀI TOÁN

Trong phần này, chúng tôi trình bày lại mô hình nổi tiếng **LT[]** và xây dựng mô hình **MLT** và mô hình cạnh trực tuyến tương ứng để mô tả quá trình lan truyền thông tin nhiều chủ đề trên **MXH**. Tiếp theo, chúng tôi định nghĩa bài toán **MMBO** trên mô hình **MLT**, chúng tôi cho thấy **MMBO** là bài toán NP-khó, tính toán hàm mục tiêu là #P-khó.

A. Mô hình ngưỡng tuyến tính LT

Trong mô hình LT, một MXH được biểu diễn bởi đồ thị $G(V,E,w)$, mỗi cạnh $(u,v) \in E$ có trọng số $w(u,v) \in [0,1]$ biểu diễn ảnh hưởng của nút u đến nút v. Nếu $(u,v) \notin E$ thì $w(u,v) = 0$, được phân bố sao cho tổng trọng số các nút u đến nút v thỏa mãn điều kiện: $\sum_{u \in N_{in}(v)} w(u,v) \leq 1$. Mỗi nút $v \in V$ có một trong 2 trạng thái: kích hoạt *(active)* hoặc không kích hoạt *(inactive)* và tập $S \subseteq V$ là tập nguồn phát tán thông tin sai lệch. Mỗi nút $v \in V$ có ngưỡng kích hoạt $\gamma_v \in [0,1]$, nếu γ_v lớn thì cần nhiều nút hàng xóm kích hoạt v, nếu γ_v bé thì nút v dễ bị kích hoạt bởi các nút hàng xóm. Gọi $\mathcal{D}^t(G,S)$ là tập các nút bị kích hoạt bởi S tại thời điểm t trên đồ thị G, quá trình lan truyền theo các bước thời gian rời rạc như sau:

- Tại thời điểm $t = 0$, tất cả các nút trong tập S đều có trạng thái kích hoạt.

- Tại thời điểm $t \geq 1$, mỗi nút v ở trạng thái không kích hoạt, sẽ bị kích hoạt nếu tổng ảnh hưởng của các nút hàng xóm tới nó vượt ngưỡng γ_v, nghĩa là: $\sum_{u \in \mathcal{D}^{t-1} \cap N_{in}(v)} w(u,v) \geq \gamma_v$. Các nút bị kích hoạt sẽ giữ nguyên trạng thái trong những bước tiếp theo. Quá trình lan truyền kết thúc khi sau mỗi bước không có nút nào được kích hoạt thêm.

B. Mô hình ngưỡng tuyến tính nhiều chủ đề có ràng buộc thời gian MLT

Một mạng xã hội được biểu diễn bởi đồ thị $G(V,E,w)$, $n = |V|$ và $m = |E|$, mỗi cạnh $(u,v) \in E$ có trọng số $w(u,v) \in [0,1]$ biểu diễn ảnh hưởng của nút u đến nút v, được phân bố sao cho tổng trọng số các nút u đến nút v thỏa mãn điều kiện: $\sum_{u \in N_{in}(v)} w(u,v) \leq 1$. Tập nguồn phát tán thông tin sai lệch q chủ đề: $S = \bigcup_{i=1}^{q} S_i$, trong đó $S_i \subseteq V, i = 1,\dots,q$ là tập các nút phát tán thông tin sai lệch chủ đề i. Mỗi nút v có q ngưỡng kích hoạt theo các chủ đề là: $\gamma_v^i \in [0,1]$ là đại diện cho mức độ quan tâm của v đối với chủ đề i. Nút v có thể có một hoặc nhiều trạng thái trong tập $q+1$ trạng thái sau: $Q = \{inactive, active\_1, active\_2, \dots, active\_q\}$. Nếu v có trạng thái *inactive* nghĩa là v không bị kích hoạt bởi bất kỳ thông tin sai lệch nào; Nếu v có trạng thái *active_i* nghĩa là v đã bị kích hoạt bởi chủ đề i. Mỗi nút v còn ảnh hưởng riêng đến nút hàng xóm theo từng chủ đề, gọi là ảnh hưởng theo chủ đề: $\theta_v^1, \theta_v^2, \dots, \theta_v^q$, với $\theta_v^i \in [0,1]$. Nghĩa là, trọng số ảnh hưởng theo từng chủ đề là: $w_i(u,v) = w(u,v).\theta_u^i$. Quá trình lan truyền thông tin diễn ra theo d bước rời rạc. Gọi $\mathcal{D}_i^t(G,S)$ là tập các nút bị kích hoạt bởi thông tin chủ đề i tại thời điểm t, ta có:

- Tại thời điểm $t = 0$, tất cả các nút trong tập S_i đều có trạng thái *active _i;*

- Tại thời điểm $1 \leq t \leq d$, các nút v chưa bị kích hoạt bởi chủ đề i sẽ có trạng thái *active_i* nếu tổng ảnh hưởng từ các nút hàng xóm tới nó vượt ngưỡng γ_v^i, nghĩa là: $\sum_{u \in N_{in}(v) \cap \mathcal{D}_i^{t-1}(G,S)} w(u,v).\theta_u^i \geq \gamma_v^i$.

Các nút bị kích hoạt sẽ giữ nguyên trạng thái kích hoạt trong những bước tiếp theo. Quá trình lan truyền kết thúc khi sau mỗi bước không có nút nào được kích hoạt hoặc đã xét hết các nút sau d bước lan truyền. Khi $q = 1$, mô hình **LT** trở thành trường hợp đặc biệt của mô hình **MLT**; khi $q > 1$, lan truyền thông tin từ tập S_i trên đồ thị G tương đương với trên đồ thị $G_i(V_i, E_i, w_i)$, $w_i(u,v) = w(u,v).\theta_u^i$. Theo [18], mô hình **LT** tương đương với mô hình cạnh trực tuyến (live edge), vì vậy MLT tương ứng với mô hình cạnh trực tuyến sau:

Từ đồ thị G, xây dựng q đồ thị $G_i(V_i, E_i, w_i)$ thỏa mãn điều kiện $w_i(u,v) = w(u,v).\theta_u^i$, $V_i \subseteq V$, $E_i \subseteq E_i$ với $i = 1,\dots,q$. Do $\theta_u^i \leq 1$, mỗi nút $v \in G_i$, ta có $\sum_{u \in N_{in}(v)} w(u,v).\theta_u^i \leq \sum_{u \in N_{in}(v)} w(u,v) \leq 1$. Suy ra, lan truyền thông tin từ tập S_i trên đồ thị G_i thỏa mãn mô hình **LT**. Hay nói cách khác, lan truyền của tập S_i trên G_i và trên G là tương đương nhau. Trên đồ thị G_i sinh ngẫu nhiên tập các đồ thị $\mathcal{G}_i$ theo mô hình cạnh trực tuyến [18], mỗi đồ thị $g \in \mathcal{G}_i$ được sinh ra bằng cách: Mỗi nút $v \in V_i$, chọn nhiều nhất một cạnh đến $(u,v), u \in N_{in}(v)$ với xác suất chọn cạnh là $p(v,g,G_i)$ được xác định như sau: $p(v,g,G_i) = w_i(u,v)$ nếu $(u,v) \in g$ và $p(v,g,G_i) = 1 - \sum_{u \in N_{in}(v)} w_i(u,v)$ nếu $(u,v) \notin g$. Như vậy g là đồ thị gồm tập nút V_g là các nút chịu ảnh hưởng bởi nguồn S_i và tập cạnh trực tuyến E_g với trọng số là $w_i(u,v)$, $u,v \in V_g$. Đặt $\Pr[g]$ là xác suất lựa chọn đồ thị g từ tập $\mathcal{G}_i$, ta có: $\Pr[g] = \prod_{v \in V_g} p(v,g,G_i)$. Gọi $N_d(g,S_i) = \{v | d_g(S_i,v) \leq d\}$ là tập các nút bị kích hoạt sau d bước lan truyền, trong đó $d_g(S_i,v)$ là khoảng cách từ S_i đến v trên đồ thị g. Gọi $\mathcal{D}_d(G_i,S_i)$ là ảnh hưởng của tập S_i trên G_i. Theo Định lý 1.1 [18], ta có:

$$\mathcal{D}_d(G_i,S_i) = \sum_{g \in \mathcal{G}_i} \Pr[g] \, |N_d(g,S_i)| \quad (1)$$

Ảnh hưởng từ tập nguồn S trên đồ thị G sau d bước lan truyền là:

$$\mathcal{D}(G,S) = \sum_{i=1}^{q} \mathcal{D}_d(G_i, S_i) \quad (2)$$

So với mô hình LT, MLT là mô hình có thể hiểu diễn lan truyền thông tin q chủ đề cũng lúc trên MXH, ngoài ra MLT có ràng buộc về thời gian d, vì vậy không gian tìm kiếm của bài toán được rút ngắn.

C. Định nghĩa bài toán

Trong bài báo này, chúng tôi đề xuất phương pháp xóa bỏ tập nút A ra khỏi đồ thị G sao cho ảnh hưởng của các tập nguồn phát tán thông tin sai lệch: $S_1, S_2, \dots, S_q$ trên đồ thị G là nhỏ nhất.

Gọi $G \odot A$ là đồ thị sau khi loại bỏ tập nút A. Theo công thức (1)(2), ảnh hưởng của S trên G sau khi loại bỏ tập A là:

$$\mathcal{D}(G \odot A, S) = \sum_{i=1}^{q} \mathcal{D}_d(G_i \odot A), S_i) \quad (3)$$

Độ giảm ảnh hưởng sau khi loại bỏ tập A được ước lượng là (hàm mục tiêu):

$$\sigma(G,S,A) = \mathcal{D}(G,S) - \mathcal{D}(G \odot A, S) \quad (4)$$

Để ngăn chặn thông tin sai lệch, công việc đầu tiên là phát hiện nguồn thông tin sai lệch. Một phương pháp phổ biến để phát hiện thông tin sai lệch là sử dụng học máy hoặc đặt giám sát. Tuy nhiên, trong bài báo này, chúng tôi giả sử rằng tập nguồn phát tán thông tin sai lệch S đã được biết trước. Giả sử chi phí để xóa bỏ nút v là $c(v)$ và tổng kinh phí để xóa bỏ tập A không vượt quá B. Bài toán **MMBO** được phát biểu như sau:

Định nghĩa 1: (MMBO) Một mạng xã hội được biểu diễn bởi đồ thị có hướng $G(V,E,w)$, có trọng số không âm, gồm n nút và m cạnh. Cho tập nguồn phát tán thông tin sai lệch q chủ đề: $S = \bigcup_{i=1}^{q} S_i$, trong đó $S_i \in V$ là tập các nút phát tán thông tin sai lệch chủ đề i. Bài toán đặt ra là tìm tập nút $A \in V$ để loại bỏ khỏi G, sao cho ảnh hưởng của thông tin sai lệch từ nguồn S trên trên G là nhỏ nhất, điều này tương đương với việc tìm tập nút $A \in V$ để loại bỏ khỏi G sao cho hàm mục tiêu $\sigma(G,S,A)$ đạt giá trị cực đại với nguồn ngân sách hạn chế B, nghĩa là $c(A) = \sum_{v \in A} c(v) \le B$.

Theo Định lý 1 trong [16], bài toán **MMBO** có độ phức tạp là NP-khó trên mô hình **MLT** ngay cả khi đồ thị là cây có gốc tại nút nguồn duy nhất. Cho $q = 1$, theo Định lý 1 trong [18] tính toán giá trị hàm $\sigma(.)$ là #P-khó ngay cả khi tập A chỉ có một nút duy nhất.

III. ĐỀ XUẤT THUẬT TOÁN

Trong phần này, đầu tiên chúng tôi chỉ ra rằng hàm mục tiêu có tính chất monotone và submodular trên mô hình **MLT**, dựa trên tính chất này, đề xuất thuật toán Tham lam **GA** đảm bảo tỷ lệ xấp xỉ $(1 - 1/\sqrt{e})$. Tiếp theo, chúng tôi rút gọn đồ thị ban đầu thành đồ thị không có chu trình theo các chủ đề và đề xuất thuật toán Tham lam tăng tốc **FGA** dựa trên vai trò lan truyền của các nút và cập nhật nhanh hàm mục tiêu, chúng tôi cho thấy tính toán hàm mục tiêu có thời gian tuyến tính và thuật toán có thể áp dụng cho đồ thị quy mô lớn.

A. Thuật toán tham lam GA

Theo Định lý 2 [18], hàm mục tiêu $\sigma(.)$ có tính chất monotone và submodular. Dựa vào tính chất này, chúng tôi đề xuất thuật toán tham lam **GA** cho tỷ lệ xấp xỉ $(1 - 1/\sqrt{e})$. Trong thuật toán **GA,** thêm dần các nút vào tập A theo kiểu ăn tham, mỗi nút v được thêm vào sao cho tỷ lệ giữa độ tăng của hàm mục tiêu $\sigma(.)$ khi loại bỏ nút v với chi phí để loại nút v ra khỏi mạng đạt giá trị lớn nhất. Quá trình kết thúc khi chi phí xóa các nút vượt mức cho phép B hoặc đã xét hết các nút sau d bước lan truyền (thuật toán 1).

Thuật toán 1: Thuật toán Tham lam GA (Greedy Algorithm)

Input: $G = (V,E)$, $w(u,v)$, *tập nguồn S, chi phí B*
Output: *Tập nút A*

1. $A \leftarrow \emptyset$; $N \leftarrow N_d(S)$;
2. **repeat**
3. $u \leftarrow argmax_{v \in V \setminus A} \frac{(\sigma(G \odot (A \cup \{u\}), S) - \sigma(G \odot A, S)}{c(u)}$
4. **if** $(c(A) + c(u) \le B)$ **then**
5. $A \leftarrow A \cup \{u\}$;
6. **end if**;
7. $N \leftarrow N \setminus \{u\}$;
8. **until** $N = \emptyset$;
9. Return A;

Tính toán chính xác giá trị hàm $\sigma(.)$ là vấn đề #P-khó. Vì vậy, thuật toán **GA** không thể áp dụng trực tiếp cho mạng xã hội thực. Để giải quyết vấn đề này, chúng tôi sử dụng phương pháp mô phỏng Monte-Carlo (MC) để ước lượng giá trị hàm $\mathcal{D}(G,S)$ và $\mathcal{D}(G \odot A,S)$. Với mỗi chủ đề $i = 1,2,..,q$, trên đồ thị G_i, tiến hành mô phỏng **MC** quá trình lan truyền thông tin từ tập S_i ngẫu nhiên T lần. Mỗi lần, chúng tôi tính tổng ảnh hưởng của các nút bị kích hoạt, sau đó lấy trung bình trên T lần mô phỏng. Cuối cùng tổng ảnh hưởng của các nút bị kích hoạt trên tất cả các chủ đề. Số lần mô phỏng T càng lớn thì ước lượng kỳ vọng số nút bị kích hoạt có độ chính xác càng cao. Thuật toán **GA** có độ phức tạp theo thời gian là $O(qTRk^2)$. R là độ phức tạp thời gian của mô phỏng **MC**, $k = |N_d(S)|$ là số nút bị kích hoạt sau d bước lan truyền. Độ phức tạp này không cho phép áp dụng cho mạng vừa và lớn, đây chính là cơ sở để chúng tôi tiếp tục đề xuất thuật toán **FGA** có thể áp dụng cho mạng lên đến hàng triệu cạnh.

B. Thuật toán Tham lam tăng tốc dựa trên DAG

Trong mục này, bài báo đề xuất hướng tiếp cận thuật toán heuristic cho bài toán **MMBO**. Từ đồ thị ban đầu, chúng tôi xây dựng các **DAG** theo q chủ đề. Sau đó ước lượng giá trị hàm ảnh hưởng $\mathcal{D}(G,S)$ trên **DAG**, cập nhật nhanh hàm mục tiêu và đề xuất thuật toán **FGA** dựa trên vai trò ảnh hưởng của các nút.

1. Xây dựng các DAG theo chủ đề

Gọi $P_d(G,u,v)$ là đường đi từ u đến v trên đồ thị $G(V,E,w)$ với độ dài (khoảng cách) d, $P_d(G,u,v) = \{u = x_1, x_2, \ldots, x_d = v\}$. Ảnh hưởng theo đường đi từ u đến v trên đồ thị G ký hiệu là $\text{Inf}(P_d(G,u,v))$ được tính như sau:

$$\text{Inf}\big(P_d(G,u,v)\big) = \prod_{i=1}^{d-1} w(x_i, x_{i+1}) \tag{5}$$

Ta có, đường đi ảnh hưởng cực đại (Maximum Influence Path - MIP) từ u đến v trên đồ thị G với độ dài không quá d, kí hiệu là $MIP_d(G,u,v)$ được tính như sau: $MIP_d(G,u,v) = argmax_{\mathcal{P} \in \{P_1(G,u,v) \cup P_2(G,u,v), \ldots, P_d(G,u,v)\}}\{\text{Inf}(\mathcal{P})\}$.

Định nghĩa 2: Cây ảnh hưởng cực đại (Maximum Influence Tree - **MIT**). Gọi $MIT_d(G,u,\beta)$ là cây ảnh hưởng cực đại theo đường đi trên đồ thị G, có nút gốc u, độ dài d và ngưỡng ảnh hưởng $\beta \in [0,1]$, β là giá trị để loại bỏ những đường đi có ảnh hưởng thấp trong quá trình lan truyền, được định nghĩa như sau:

$$MIT_d(G,u,\beta) = \bigcup_{v \in V, \inf(MIP_d(G,u,v)) \geq \beta} MIP_d(G,u,v) \tag{6}$$

Gọi D_i là **DAG** được xây dựng từ đồ thị $G_i(V,E,w_i)$ với $w_i(u,v) = w(u,v).\theta_i^u$. Để xây dựng D_i, đầu tiên chúng tôi gộp các nút nguồn S_i thành một nút nguồn duy nhất H_i theo thuật toán 3 trong [16]. Trên đồ thị G_i, xây dựng cây tối đa ảnh hưởng theo từng chủ đề $\text{MIT}(X_i, Y_i, \beta_i)$ bằng thuật toán Dijkstra với trọng số cạnh là $-\log w_i(u,v)$, với X là tập nút và Y là tập cạnh, β_i là ngưỡng lan truyền ảnh hưởng theo chủ đề i. Cây được tạo ra gồm nút gốc H_i và các đường đi có ảnh hưởng lớn hơn ngưỡng lan truyền. Tiếp theo, trên cây $\text{MIT}(X_i, Y_i, \beta_i)$ thêm dần các cạnh (u,v) vào cây sao cho độ cao từ H_i đến u nhỏ hơn độ cao từ H_i đến v và $(u,v) \in E$. Kết quả thu được là **DAG** D_i. (thuật toán 2).

Thuật toán 2: Xây dựng đồ thị không có chu trình D_i theo chủ đề i

Input: $G_i(V,E,w_i)$, ngưỡng ảnh hưởng β_i, nút nguồn H_i
Output: DAG D_i
1. $D_i = MIT_d(G_i, H_i, \beta_i)$;
2. **foreach** adge $(u,v) \in G_i and\ (u,v) \neq D_i$ **do**
3. **if** $hight(H_i,u) \leq hight(H_i,v)$ **then**
4. $D_i = D_i \cup \{(u,v)\}$
5. **end if**.
6. **end for**
7. **Return** D_i

2. Ước lượng giá trị của hàm ảnh hưởng $\mathcal{D}(\text{G}, \text{S})$ trên DAG

Tính toán $\sigma(.)$ trên đồ thị G là #P-khó, chúng tôi cho thấy tính toán hàm mục tiêu $\sigma(.)$ trên **DAG** có thời gian đa thức. Theo công thức 6. Ảnh hưởng từ nút nguồn H_i đến nút v trên D_i (ký hiệu là $f_i(H_i,v)$):

$$f_i(H_i,v) = \sum_{\mathcal{P} \in P(D_i,H_i,v)} \text{Inf}(\mathcal{P}) \tag{7}$$

Trong đó $P(D_i,H_i,v)$ là tập đường đi từ H_i đến v trên D_i với $\forall v \in D_i$.

Ảnh hưởng của H_i trên D_i được tính bằng tổng ảnh hưởng của H_i đến tất cả các nút, $u \in D_i$:

$$\mathcal{D}_i(D_i,H_i) = \sum_{v \in D_i} f_i(H_i,v) = \sum_{v \in D_i} \sum_{\mathcal{P} \in P(D_i,H_i,v)} \text{Inf}(\mathcal{P}) \tag{8}$$

Để cập nhật giá trị hàm $f_i(H_i,u)$**,** trong Thuật toán 3, sắp xếp theo thứ tự Topo các nút của D_i bắt đầu từ nút gốc H_i theo phương pháp duyệt sâu (dòng 2) để đảm bảo rằng khi tính $f_i(H_i,u)$ thì các nút trước nó đã được tính. Từ dòng 3

đến dòng 7, trả về giá trị hàm $f_i(H_i, u)$ bằng tổng ảnh hưởng các đường đi từ H_i đến u trên D_i. Thuật toán 4, tính giá trị ảnh hưởng của H_i đến trên D_i.

Thuật toán 3: Tính ảnh hưởng từ nút nguồn đến nút $u \in D_i$

Input: DAG D_i, nút nguồn H_i, $u \in D_i$
Output: $f_i(H_i, u)$

1. $f_i(H_i, H_i) \leftarrow 1$, $f_{in}(H_i, u) \leftarrow 0$;
2. Topologically sort all nodes from H_i in D_i into a sequence List based on DFS
3. **foreach** $u \in List$ *(from the first to last)* **do**
4. **foreach** $v \in N_{in}(u)$ **do**
5. $f_{in}(H_i, v) \leftarrow \big(f_{in}(H_i, v) + f_{in}(H_i, u).w_i(u, v)\big)$;
6. **end for**
7. **end for**
8. **Retturn** $f_i(H_i, u)$.

Thuật toán 4: Tính ảnh hưởng từ nút nguồn trên DAG D_i

Input: DAG D_i, nút nguồn H_i,
Output: $\mathcal{D}_i(D_i, H_i)$

1. $x \leftarrow 0$;
2. **foreach** $u \in D_i$ **do**
3. call $f_i(H_i, u)$ (alg 3)
4. $x \leftarrow x + f_i(H_i, u)$;
5. **end for**
6. **Retturn** x.

Ảnh hưởng của tập nguồn S trên đồ thị G được ước lượng bằng tổng ảnh hưởng của các nút nguồn H_i trên DAG D_i, $i = 1, \ldots, q$, được tính như sau:

$$\mathcal{D}(G, S) \approx \sum_{i=1}^{q} \mathcal{D}_i(G_i, H_i) \quad (9)$$

Suy ra, ảnh hưởng S trên G sau khi loại bỏ tập nút A là: $\mathcal{D}(G \odot A, S) = \sum_{i=1}^{q} \mathcal{D}(G_i \odot A, S_i)$ (10)

Từ công thức (3), (9) (10) và thuật toán 3, ta nhận thấy, tính toán $\sigma(.)$ được thực hiện trong thời gian đa thức.

3. Ước lượng vai trò ảnh hưởng của các nút trên **DAG**

Vai trò ảnh hưởng của v dựa vào ảnh hưởng từ nút nguồn đến v và ảnh hưởng từ v đến các nút khác trên **DAG** D. Vai trò ảnh hưởng càng lớn thì ảnh hưởng của v trên D càng cao. Vì vậy, **FGA** sử dụng đại lượng này có thể chọn các nút đưa vào tập lời giải A, thay thế đại lượng $\delta(v)$ trong thuật toán **GA**.

Gọi $f_i(u)$ là ảnh hưởng từ nút u đến các nút khác trên D_i, ta có: $f_i(u) = \sum_{v \in D_i} \sum_{\mathcal{P} \in P(D_i, u, v)} \text{Inf}(\mathcal{P})$

Ký hiệu $R_i(u)$ là vai trò ảnh hưởng của nút u trên D_i được tính là: $R_i(u) = f_i(H_i, u).f_i(u)$ (Thuật toán 5)

Vai trò ảnh hưởng của u đối với q của đề được ước lượng là: $(u) = \frac{1}{q}\sum_{i=1}^{q} R_i(u)$ (11)

Thuật toán 5: Ước lượng vai trò ảnh hưởng của các nút trên DAG D_i

Input: DAG D_i, nút nguồn H_i
Output: $R_i(u), \forall u \in D_i$

1. $f_{out}(u) = f_{out}(H_i) \leftarrow 1$;
2. Topologically sort all nodes from H_i in D_i into a sequence List based on DFS;
3. **foreach** $u \in D_i$ **do**
4. call $f_i(H_i, u)$ (alg 5)
5. **foreach** $u \in List$ *(from the last to first)* **do**
6. **foreach** $v \in N_{out}(u)$ **do**
7. $f_{out}(v) \leftarrow (f_{out}(v) + f_{out}(u).w_i(u, v))$;
8. **Retturn** $f_i(H_i, u).f_i(u)$.

4. Thuật toán Tham lam tăng tốc FGA

Thuật toán **FGA** (Thuật toán 6) là sự kết hợp giữa việc xây dựng các **DAG** theo chủ đề và thuật toán **GA**. Trong thuật toán **FGA**, hàm $\delta(u)$ được thay thế bằng vai trò ảnh hưởng $R(u)$. Thuật toán được chia làm 3 phần như sau:

Đầu tiên, dựng q đồ thị G_i, $w_i(u,v) = w(u,v).\theta_u^i$ từ đồ thị G. Trên đồ thị G_i, gộp các nút trong S_i thành một nút nguồn duy nhất H_i (thuật toán 3 trong [17]). Rút gọn G_i thành **DAG** D_i, $i = 1,2,\ldots,q$ (thuật toán 4), tính toán vai trò ảnh hưởng của tất cả các nút và giá trị ảnh hưởng của H_i trên D_i (dòng 1-9), đồng thời xây dựng tập nút ứng viên U là các nút có thể chọn để loại bỏ ra khỏi G, tính tổng ảnh hưởng của tập nguồn S trên đồ thị G theo công thức 1 (dòng 10). Thứ hai, chọn tập A là một nút duy nhất v_{max} có vai trò ảnh hưởng lớn nhất trên G mà chi phí loại bỏ $v_{\max}$ nhỏ hơn hoặc bằng B (dòng 11); Chọn nút u thỏa mãn điều kiện $R(u)/c(u)$ lớn nhất và $c(u) + c(A') \leq B$ đưa vào tập A' (dòng 16), cập nhật lại đồ thị G_i và xây dựng D_i từ các đồ thị G_i mới (G^{cur}), từ đó cập nhật vai trò ảnh hưởng mới của các nút $u \in D_i$. Quá trình tiếp tục cho đến khi tập $U = \emptyset$ hoặc $c(A') > B$. Cuối cùng, thuật toán so sánh giá trị ảnh hưởng của tập S trên đồ thị G đối với phương án loại bỏ tập A' và loại bỏ tập A để chọn phương án tốt nhất (dòng 31 đến dòng 35).

Thuật toán 6: Thuật toán tham lam tăng tốc FGA (Fast Greedy Algorithm)

Input: $G(V,E,w), S = \{S_1, S_2, \ldots, S_q\}, B > 0, \beta = \{\beta_1, \beta_2, \ldots, \beta_q\}, d$
Output: Tập nút A

1. Construct q graph G_i; // $(w_i(u,v) = w(u,v).\theta_u^i$
2. **For each** $G_i | i = 1..q$ **do**
3. $(G'_i, H_i) \leftarrow Merge(G_i, S_i)$
4. $G_i^{cur} \leftarrow G'_i$
5. Construct graph $D_i = DAG(G'_i, H_i, \beta_i)$, (thuật toán 4)
6. Calculate $R_i(u), \forall u \in D_i$, (thuật toán 5)
7. Calculate $R(u), \forall u \in D_i | i = 1..q$, (công thức 11)
8. $U \leftarrow U \cup \{u\}, \forall u \in D_i | i = 1..q$
9. **end for**
10. Calculate $\mathcal{D}(G,S)$ (công thức 9)
11. $v_{max} \leftarrow argmax_{v \in U} R(v)$
12. **Repeat**
13. $c_{min} \leftarrow argmin_{v \in U} c(v)$
14. **if** $(c_{min} + c(A')) \geq B$ **then** break;
15. **for each** $u \in U$ **do**
16. **if** $(c(u) + c(A') \leq B)$ **then**
17. $u_{max} \leftarrow argmax_{u \in U, c(A_1 \cup \{v\}) \leq B} R(u)/c(u)$
18. **else**
19. $U \leftarrow U \backslash \{u\}$
20. **end if**
21. **end for**
22. $U \leftarrow U \backslash \{u_{max}\}$
23. $A' \leftarrow A' \cup \{u_{max}\}$
24. **for each** $G_i | i = 1..q$ **do**
25. $G_i^{cur} \leftarrow G_i^{cur} \backslash \{u_{max}\}$
26. Construct graph $D_i = DAG(G_i^{cur}, H_i, \beta_i)$
27. Calculate $R_i(u), R(u), \forall u \in D_i | i = 1..q$,
28. **end for**
29. **Until** $U = \emptyset$ or $c(A') \geq B$;
30. $x \leftarrow \mathcal{D}(G \odot A', S)$; $y \leftarrow \mathcal{D}(G \odot v_{max}, S)$;
31. **Return** A_1 if $(x > y)$ else v_{max}.

Độ phức tạp của **FGA** được tính như sau: Gọi n_i, m_i là số nút và số cạnh của D_i. Tính toán $MIT_d(G_i, A, \theta)$ thực hiện trong thời gian $O(m_i + n_i \, log n_i)$. Do vậy việc dựng **DAG** D_i từ G_i là $O(m_i + n_i \, log n_i)$ (dòng 5). Tính toán hàm $R_i(u), \forall u \in D_i$ sử dụng thuật toán 5 có độ phức tạp là $O(m_i + n_i)$ (dòng 6). Với vòng lặp repeat (dòng 6-32), gọi k_i số vòng lặp. Với mỗi vòng lặp, việc dựng **DAG** và tính toán $r(\cdot)$ thực hiện trong thời gian $O(m_i + n_i) + O(m_i + n_i \, log n_i)$. Việc so sánh kết quả cuối cùng thực tốn chi phí là $O(m_i + n_i)$. Từ các phân tích trên, với q chủ đề thông tin sai lệch ta có độ phức tạp của thuật toán là $O(qk_i(mm_i + n_i \, log n_i))$.

IV. KẾT QUẢ THỰC NGHIỆM

Trong phần này, chúng tôi trình bày cách thức tiến hành thực nghiệm và kết quả thực nghiệm nhằm đánh giá hiệu quả của thuật toán **GA** và **FGA**. Mặc dù đã có nhiều công bố liên quan đến bài toán **IB**, tuy nhiên tác giả chưa tìm mấy các công bố về lớp bài toán **IB** nhiều chủ đề. Vì vậy, hiệu quả của thuật toán chúng tôi đề xuất được so sánh với các thuật toán cơ sở là Degree và Random. Chúng tôi tiến hành chạy thực nghiệm trên các 03 bộ dữ liệu của mạng thực tế là NetHepPh[20], Epinions[21], Amazon[22], được lấy từ nguồn [http://snap.stanford.edu/data/]. Các thuật toán được cài đặt bằng ngôn ngữ lập trình Python trên máy tính có cấu hình: CPU Intel Core i7 - 8550U 1.8Ghz, RAM 8GB DDR4 2400MHz, hệ điều hành Linux.

A. Cài đặt thực nghiệm

Vì khó có thể xác định chính xác trọng số ảnh hưởng của u đối với v, nên chúng tôi căn cứ trên thực nghiệm của các nghiên cứu [9,17,18]: Mỗi cạnh (u,v) có trọng số ảnh hưởng là: $w(u,v) = 1/|N_{in}(v)|$, nghĩa là: $\sum_{u \in N_{in}(v)} w(u,v) = 1$; Chi phí loại bỏ nút $c(v)$, $v \in V$ được khởi tạo ngẫu nhiên trong khoảng [1.0, 3.0]; Bộ nguồn phát tán thông tin sai lệch S gồm có 03 chủ đề ($q = 3$), mỗi chủ đề được lấy ngẫu nhiên 100 nút: $|S_1| = 100, |S_2| = 100, |S_3| = 100$. Ngân sách $B = 100$, số bước lan truyền thông tin $d = 6$. Đối với thuật toán **GA**, ngưỡng kích hoạt được lấy ngẫu nhiên $\gamma_v^i \in [0,1]$ và phương pháp mô phỏng MC được thực hiện 10.000 lần để tính xấp xỉ trung bình mẫu; Đối với thuật toán **FGA**, ngưỡng lan truyền $\beta = 1/230$ [19] cho tất cả các chủ đề.

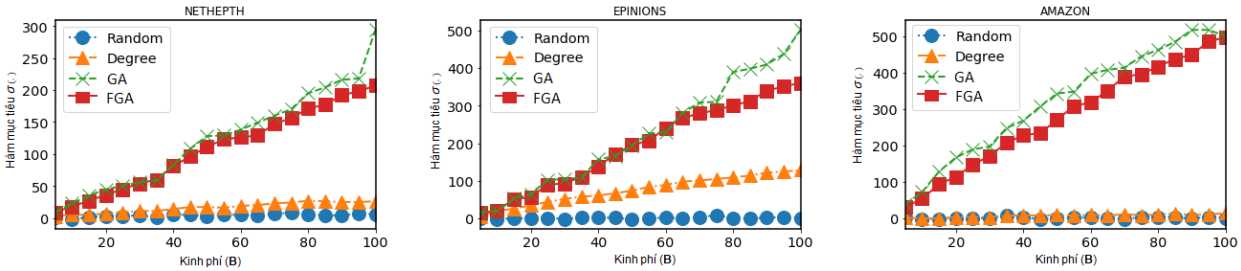
Bảng 1. Bảng dữ liệu thực nghiệm

	Số nút	**Số cạnh**	**Bậc lớn nhất**	**Bậc trung bình**
NetHept[20]	15K	31K	64	4.12
Epinions[21]	76K	509K	3079	11
Amazon[22]	262K	1.2M	425	9.4

B. Đánh giá kết quả

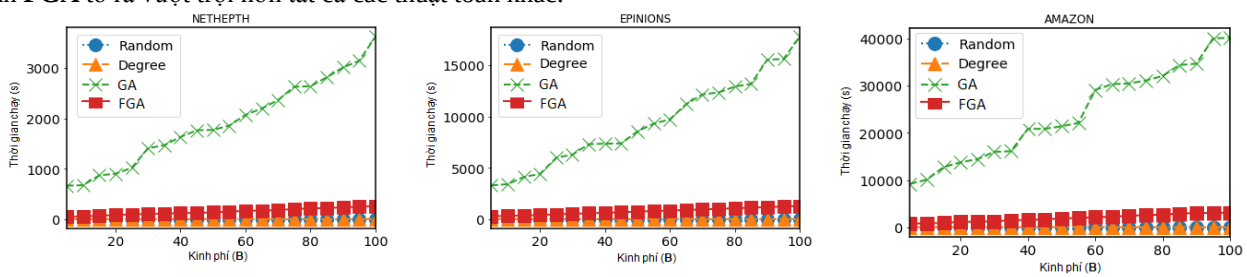
Các thuật toán được đánh giá dựa trên hai tiêu chí: 1) Chất lượng lời giải (giá trị của hàm mục tiêu); 2) Thời gian chạy của thuật toán (tính bằng giây). Chúng tôi so sánh thuật toán **GA** và thuật toán **FGA** với hai thuật toán cơ bản là Degree và Random. Thuật toán Degree: Chọn nút có bậc lớn nhất thêm vào tập A cho đến khi tổng chi phí xóa bỏ các nút vượt quá B; Thuật toán Random: Chọn ngẫu nhiên các nút trong giới hạn nguồn ngân sách B.

1. So sánh chất lượng lời giải: Hình 1 cho thấy, hiệu suất thuật toán **GA** và **FGA** cao hơn rất nhiều so với thuật toán cơ sở là Degree và Random. Ngân sách B càng tăng và mạng càng lớn thì sự vượt trội càng rõ ràng hơn. Với bộ NETHEPTH, $B = 20$, hiệu suất của **GA, FGA** gấp 24 lần so với Degree và Random; với bộ Amazon, $B = 100$, hiệu suất thuật toán **GA và FGA** cao hơn Degree và Random hơn 500 lần. So sánh giữa **GA** và **FGA** cho thấy, với chi phí B từ 0 đến 40, **FGA** và **GA** xấp xỉ bằng nhau, với $B > 40$ hiệu suất **GA** luôn cao hơn hoặc bằng **FGA**, tuy nhiên độ chệnh lệch này là không đáng kể, vì vậy có thể kết luận **FGA** có hiệu suất gần bằng với **GA** và gần đạt xấp xỉ 63,3 %.



Hình 1. So sánh hiệu suất của các thuật toán

2. So sánh thời gian chạy của thuật toán: Hình 2 cho thấy, đối với 03 bộ dữ liệu thực nghiệm, thời gian chạy của thuật toán Degree, Random là thấp nhất do tính đơn giản của thuật toán. Thuật toán **FGA**, với tính toán hàm mục tiêu dựa trên **DAG** có thời gian chạy xấp xỉ thuật toán Degree và Random. **GA** tỏ ra là thuật toán có thời gian chạy cao nhất. So sánh thời gian chạy giữa **GA** và **FGA** ta thấy khoảng cách này là quá xa. Với bộ Amazon, $B = 100$, thời gian chạy của **GA** là 60.000 giây, trong khi bộ dữ liệu **FGA** chỉ mất 10 giây để thực hiện. Vì vậy, đối với mạng lớn, thuật toán **FGA** tỏ ra vượt trội hơn tất cả các thuật toán khác.



Hình 2. So sánh thời gian chạy của các thuật toán

V. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất giải quyết bài toán ngăn chặn thông tin sai lệch đa chủ đề trên mạng xã hội trực tuyến **MMBO**, có xét ràng buộc thời gian và chi phí. Chúng tôi chứng minh bài toán **MMBO** có độ phức tạp NP-khó và tính toán hàm mục tiêu trên đồ thị G là #P-khó. Để mô tả quá trình lan truyền thông tin nhiều chủ đề trên **MXH,** chúng tôi xây dựng mô hình **MLT** và mô hình cạnh trực tuyến tương ứng. Trên mô hình này, chúng tôi đề xuất thuật toán **GA** theo chiến lược tham lam và thuật toán **FGA** là sự kết hợp giữa **GA** với vai trò ảnh hưởng của các nút và cập

nhật nhanh hàm mục tiêu trên **DAG**, thuật toán được đề xuất cho hiệu suất cao hơn các thuật toán cơ sở là Degree và Random. Do hàm mục tiêu có tính chất submodula nên hiệu suất **GA** đạt xấp xỉ $(1 - 1/\sqrt{e})$ và vượt trội hơn **FGA**, tuy nhiên độ vượt trội này là không lớn. Xét về thời thời gian thì **GA** không thể áp dụng cho các mạng xã hội thực. Thuật toán **FGA** không đạt đến xấp xỉ như **GA**, nhưng có thời gian chạy đa thức, nên có thể áp dụng **FGA** cho các mạng lên đến hàng trăm nghìn đỉnh và hàng triệu cạnh. Thời gian tới, chúng tôi tiếp tục nghiên cứu nề nâng cao hiệu suất của **FGA** và cải tiến thuật toán **GA** để có thể áp dụng cho các mạng xã hội thực.

VI. TÀI LIỆU THAM KHẢO

[1] Domm P (2013). "False rumor of explosion at white house causes stocks to briefly plunge"; AP confirms its Twitter feed was hacked CNBC. CNBC http://www.cnbc.com/id/100646197. Accessed 21 July 2019.

[2] AllcottH, Gentzkow M (2016). "Social media and fake news in the 2016 election". Stanford Webhttps://web.stanford.edu/~gentzkow/research/fakenews.pdf. Accessed 21 July 2019.

[3] M. Kimura, K. Saito, and H. Motoda. "Solving the contamination minimization problem on networks for the linear threshold model," in PRICAI 2008, Hanoi, Vietnam, December 15-19, 2008. Proceedings, pp. 977-984, 2008.

[4] M. Kimura, K. Saito, and H.Motoda, "Blocking links to minimize contamination spread in a social network," ACM TKDD, Vol. 3, No. 2, pp. 9:1-9:23, 2009.

[5] Y. Zhang and A. Prakash, "Scalable vaccine distribution in large graphs given uncertain data," in the 23rd ACM CIKM 2014, Shanghai, China, November 3-7, 2014, pp. 1719-1728, 2014.

[6] Y. Zhang and B. A. Prakash, "Data-aware vaccine allocation over large networks," TKDD, Vol. 10, No. 2, pp. 20:1-20:32, 2015.

[7] Y. Zhang, A. Adiga, S. Saha, A. Vullikanti, and B. A. Prakash, "Near-optimal algorithms for controlling propagation at group scale on networks," IEEE Trans. Knowl. Data Eng., Vol. 28, No. 12, pp. 3339-3352, 2016.

[8] X. He, G. Song, W. Chen, and Q. Jiang. "Influence blocking maximization in social networks under the competitive linear threshold model". In Proceedings of the 12th SIAM International conferenceon data mining, anaheim, California, USA, April 26-28, 2012, pp. 463-474, 2012.

[9] C. Song and M. Lee, "Temporal influence blocking: Minimizing the effect of misinformation in social networks," in In 33rd IEEE International Conference on Data Engineering, ICDE2017, SanDiego, CA, USA, April 19-22, pp. 847-858, 2017.

[10] N. P. Nguyen and M. T. Thai, "Analysis of misinformation containment in online social networks," in Computer Networks, 57(10): pp. 2133-2146, 2013.

[11] J. Fan, J. Qiu, Y. Li, Q. Meng, D. Zhang, G. Li, K. Tan, and X. Du, "OCTOPUS: an online topic-aware influence analysis system for social networks," in The 34th IEEE ICDE, Paris, France, April 16-19, pp. 1569-1572, 2018.

[12] Y. Li, D. Zhang, and K. Tan, "Real-time targeted influence maximization for online advertisements," PVLDB, Vol. 8, No. 10, pp. 1070-1081, 2015.

[13] H. Sun, X. Gao, G. Chen, J. Gu, and Y. Wang, "Multiple influencemaximization in social networks," in Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, IMCOM'16, ACM, New York, NY, USA, pp. 44:1-44:8, 2016.

[14] Y. Li, J. Fan, G. Ovchinnikov, and P. Karras, "Maximizing multifacetednetwork influence," in 2019 IEEE 35th International Conference on DataEngineering (ICDE), pp. 446-457, 2019.

[15] X. Tang, Q. Miao, S. Yu, and Y. Quan, "A data-based approach to discov-ering multi-topic influential leaders," in PLOS ONE 11, Volume 7, 2016,http://dx.doi.org/10.1371/journal.pone.0158855, 2016.

[16] Dung V. Pham, Giang L. Nguyen, Tu N. Nguyen, Canh V. Pham, and Anh V. Nguyen, "Multi-topic misinformation blocking with budget constraint on online social networks", IEEE access, Q1, pp. 2020.

[17] D. Kempe, J. M. Kleinberg, and ´E. Tardos, "Maximizing the spread of influence through a social network," in The Ninth ACM SIGKDD, Washington, DC, USA, August 24 - 27, pp. 137-146, 2003.

[18] E. B. Khalil, B. N. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in the 20th ACM SIGKDD, KDD '14, New York, NY, USA - August 24 - 27, pp. 1226-1235, 2014.

[19] W. Chen, Y. Yuan and L. Zhang, "Scalable Influence Maximization in Social Networks under the Linear Threshold Model," 2010 IEEE International Conference on Data Mining, Sydney, NSW, pp. 88-97, doi: 10.1109/ICDM.2010.118, 2010.

[20] J. Leskovec, M. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in The Eleventh ACM SIGKDD, Chicago, Illinois, USA, August 21-24, pp. 177-187, 2005.

[21] M. Richardson, R. Agrawal, and P. M. Domingos, "Trust management for the semantic web," in ISWC 2003, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings, pp. 351-368, 2003.

[22] "Amazon product co-purchasing network, march 02 2003," http://snap. stanford. edu/ data/ amazon0302. html.

MULTI-TOPIC MISINFORMATION BLOCKING ON ONLINE SOCIAL NETWORKS

Pham Van Dung, Nguyen Thi Tuyet Trinh, Nguyen Viet Anh

ABSTRACT: *In recent years, Online Social Network (OSN) is becoming the popular means of communication in the world. In addition to the benefits that social media brings, it also allows the rapid spread of misinformation, which greatly affects users. Limit order to prevent the effect of misinformation on OSN. In this paper, we propose a solution to find a node set that, when removing them from the network, minimizes the effect of misinformation on the OSN with time and cost. Proven problem is NP-hard even in cases where OSN is a tree root at the node that spreads the only misinformation and calculates the target function as # P-hard even if the set needs to be deleted with only one node. We also prove the target function of monotone and submodular properties, based on this property we propose the Greedy algorithm (GA) for the approximate ratio is . We continue to propose Fast Greedy Algorithm (FGA) based on no cycle graph and the role influence of nodes, we show that target function calculation is done in polynomial time and algorithm is The proposal is applicable to networks of up to millions of edges. Experiments conducted on real-world datasets show efficiency and effectiveness of our proposed algorithm in comparison with other state-of-the-art algorithm.*

Keyword: *Optimization, Social networks, information diffusion, misinformation blocking.*