ĐẠI HỌC THÁI NGUYÊN
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG THÁI NGUYÊN

MỘT SỐ THUẬT TOÁN TÌM CORE VÀ ỨNG DỤNG TRONG PHÂN TÍCH MẠNG XÃ HỘI

ĐỖ KHẮC HOÀN

THÁI NGUYÊN 2017

LỜI CAM ĐOAN

Tôi xin cam đoan: Luận văn thạc sỹ Khoa học máy tính "**Một số thuật toán tìm core và ứng dụng trong phân tích mạng xã hội**" do tôi thực hiện và trình bày dưới sự hướng dẫn của **TS. Trương Hà Hải,** Trường Đại học Công nghệ Thông tin và Truyền thông – Đại học Thái Nguyên là công trình nghiên cứu hoàn toàn trung thực, không vi phạm bất cứ điều gì trong Luật Sở hữu trí tuệ và Pháp luật Việt Nam. Nếu sai, tôi hoàn toàn chịu trách nhiệm trước Pháp luật.

Tất cả các bài báo, khóa luận, tài liệu, công cụ phần mềm của các tác giả khác được sử dụng lại trong khóa luận này đều được chỉ dẫn tường minh về tác giả và đều có trong danh mục tài liệu tham khảo.

Thái Nguyên, ngày tháng năm 2017

Tác giả

Đỗ Khắc Hoàn

LỜI CẢM ƠN

Trước tiên, tác giả xin gửi lời cảm ơn tới tất cả quý thầy cô đã giảng dạy và quản lý chương trình Cao học chuyên ngành Khoa học máy tính của Trường Đại học Công nghệ thông tin và Truyền thông. Các thầy cô đã truyền đạt cho tác giả kiến thức chuyên ngành khoa học máy tính để tác giả làm cơ sở hoàn thành luận văn này.

Tác giả xin gửi lời cảm ơn chân thành nhất đến **TS. Trương Hà Hải**, Cô đã định hướng đề tài và tận tình hướng dẫn, chỉ bảo tác giả trong suốt quá trình thực hiện luận văn cao học này.

Sau cùng, tác giả xin dành tình cảm đặc biệt và biết ơn tới gia đình và người thân của tác giả, những người đã ủng hộ, khuyến khích và hỗ trợ tác giả rất nhiều trong quá trình học tập, nghiên cứu cũng như thực hiện luận văn này.

Do thời gian có hạn và kinh nghiệm nghiên cứu khoa học chưa nhiều nên luận văn còn nhiều thiếu sót, rất mong được sự đóng góp ý kiến của quý thầy cô và các bạn học viên để đề tài đạt kết quả cao.

Xin chân thành cảm ơn!

Thái Nguyên, ngày tháng năm 2017

Tác giả

Đỗ Khắc Hoàn

MỤC LỤC

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

Từ thế kỷ 20, lý thuyết đồ thị trở nên rất phổ biến vì ứng dụng rộng rãi của nó trong rất nhiều khía cạnh của đời sống như sinh học, xã hội học, công nghệ thông tin, mạng thông tin,…Vào năm 1930 bài toán phân tích mạng xã hội ra đời và trở thành chủ đề quan trọng nhất trong xã hội học. Trong thời đại bùng nổ thông tin hiện nay, số lượng và kích thước các mạng xã hội trực tuyến tăng lên không ngừng. Vì vậy, việc dự đoán liên kết trong mạng xã hội trực tuyến là một nhu cầu bức thiết trong thời điểm hiện nay, vì ứng dụng quan trọng của cộng đồng trong các lĩnh vực đời sống xã hội, như khoa học máy tính, sinh học, …

Mạng xã hội là một mô hình mạng có tính chất xã hội được cấu tạo bởi các đỉnh và các cung, các đỉnh liên kết với nhau bởi một hoặc nhiều cung, thể hiện mối quan hệ cụ thể. Mỗi đỉnh là một thực thể trong mạng, thực thể này có thể là một cá nhân, một tổ chức hay một quốc gia bất kỳ… Các thực thể trong mạng tương tác với nhau thông qua các liên kết. Các liên kết này có thể là quan hệ bạn bè, đồng nghiệp, cũng có thể là các quan hệ đối đầu thù địch hay các trao đổi tài chính, giao dịch…

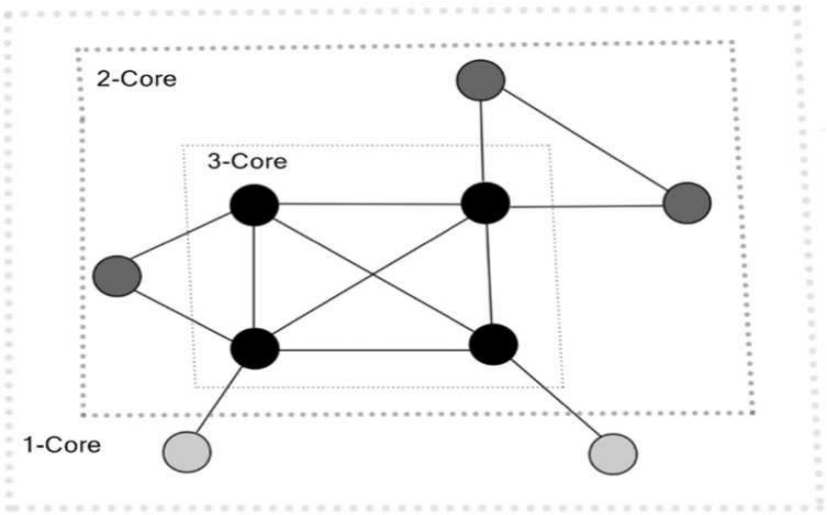
Nhu cầu phân tích mạng xã hội đã được bắt đầu từ rất sớm từ những năm 1930 và ngày càng trở thành chủ đề quan trọng. Đặc biệt với sự phát triển hiện nay của mạng xã hội đã sản sinh ra một khối lượng dữ liệu khổng lồ, vì vậy bài toán phân tích mạng xã hội trở thành bài toán phân tích mạng trong miền dữ liệu lớn. Đây là một bài toán khó và nhận được nhiều sự quan tâm của các nhà khoa học hiện nay.

Một trong những mối quan tâm lớn của mạng xã hội là phân tích và xác định các nhóm con gắn kết (cohesive groups) trong mạng. Một số khái niệm đã được đưa ra để mô tả tính kết hợp giữa các nhóm này, đó là: *cliques, n–cliques, n–clans, n–clubs, k–plexes, k–cores,…* Bài toán tìm các nhóm kết hợp là bài toán *NP- hard*. Khái niệm k-lõi (*k-core*) được Seidman đưa ra vào năm 1983 [7] là một cách phân tách mạng lớn thành các mạng nhỏ hơn để dễ xử lý. Các thuật toán *k-core* đưa ra để tìm các nhóm nhỏ trong mạng và phân chúng ra thành những mạng nhỏ hơn, đến khi đạt được kết quả là các nhóm nhỏ nhất. Đã có

nhiều thuật toán được đề xuất để tìm *k-core*, trong đó có những thuật toán khá hiệu quả, có độ phức tạp đa thức [3, 4, 5, 6, 7].

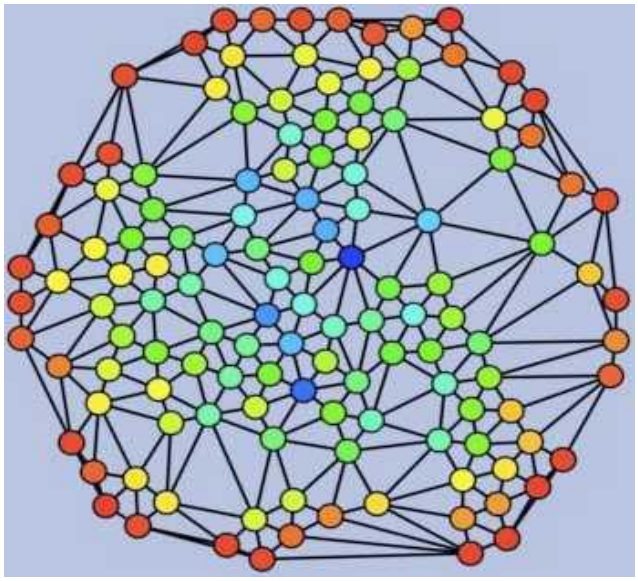
Với những ứng dụng thực tế rất ý nghĩa của mạng xã hội, trong thời đại bùng nổ thông tin hiện nay, số lượng và kích thước các mạng xã hội trực tuyến tăng lên không ngừng. Vì vậy, việc phân tích mạng xã hội là một nhu cầu bức thiết trong thời điểm hiện nay, vì ứng dụng quan trọng của cộng đồng trong các lĩnh vực của đời sống xã hội, như khoa học máy tính, sinh học, kinh tế, chính trị,…Nội dung chính của luận văn này là nghiên cứu một số thuật toán tìm *k-core* và ứng dụng của *k-core* trong phân tích mạng xã hội, từ đó có thể áp dụng giải một bài toán trong thực tế.

Thuật toán về *k-core* đưa ra để phân tích cấu trúc tính toán các nhóm nhỏ trong mạng và phân chúng ra thành những mạng nhỏ hơn, đến khi đạt được kết quả là các nhóm nhỏ nhất. Nhưng giữa các nhóm trong mạng vẫn có mối liên kết chặt chẽ với nhau thông qua các nút mạng của nhóm. Ngoài ra thuật toán về k-core được sử dụng để mô tả lưới của một mạng lưới, bằng cách tìm mật độ mạng trực tiếp, chuỗi các đỉnh trong tuần tự có thể xác định bởi số lượng các nút của đồ thị đó.



Hình 1: Mô hình k-core phân rã thành những k-core nhỏ khác nhau trong phác thảo một đồ thị nhỏ [7].

Xác định các khái niệm về k-core một số phương pháp tìm kiếm đơn giản dễ thực hiện và tính toán được dựa trên kiến thức của các đỉnh trong đồ thị, thuật toán k-core địa phương, thuật toán Trie Data structure, thuật toán phân hủy. Cho thấy được mối quan hệ bài toán với việc tìm mạng xã hội và thuật toán k-core. Kết quả đạt được cho thấy sự hiệu quả của thuật toán và cấu trúc đồ thị với ứng dụng mạng xã hội.



Hình 2: Độ phân rã K-core trong phân tích mạng xã hội [9].

Luận văn tập trung tìm hiểu tổng quan các kiến thức có liên quan, các cơ sở lý thuyết như: Cấu trúc mạng, liên kết mạng xã hội. Một số thuật toán tìm core, ứng dụng trong phân tích mạng xã hội.

Luận văn được trình bày thành 3 phần bao gồm: phần mở đầu, phần nội dung và phần kết luận.

Phần mở đầu:

Giới thiệu khái quát về đề tài, mục tiêu, đối tượng, phạm vi nghiên cứu, ý nghĩa khoa học và xã hội mang lại thông qua việc giải quyết các vấn đề được nêu trong đề tài.

Phần nội dung:

Chương 1: Cơ sở lý thuyết về đồ thị và mạng xã hội

Nội dung cơ bản của chương: Trình bày một số kiến thức tổng quan liên quan đến nội dung đề tài.

Chương 2: Một số thuật toán nhanh tìm k-core trong mạng xã hội

Tìm hiểu một số thuật toán tìm Cores trong phân tích mạng xã hội, mổ tả thuật toán, đánh giá độ phức tạp của thuật toán.

Chương 3. Ứng dụng của core trong phân tích mạng xã hội

Nội dung cơ bản trong chương này: Tìm hiểu một số ứng dụng của core trong phân tích mạng xã hội và xây dựng chương trình ứng dụng.

Phần kết luận:

Trình bày kết quả mà luận văn đạt được và phương hướng đề xuất.

CHƯƠNG 1.
CƠ SỞ LÝ THUYẾT ĐỒ THỊ VÀ MẠNG XÃ HỘI

Phân tích mạng xã hội được xem là các mối quan hệ xã hội về lý thuyết mạng lưới bao gồm các nút và các mối quan hệ (còn gọi là các cạnh, liên kết, hoặc kết nối). Nút là các cá nhân trong mạng lưới, và các mối quan hệ là những mối liên kết với các cá nhân. Kết quả là các cấu trúc dựa trên đồ thị rất phức tạp.

Nội dung cơ bản của chương trình bày các khái niệm cơ sở về đồ thị, các loại đồ thị, một số khái niệm về phân tích mạng xã hội cũng như khái niệm về thuật toán tìm core để làm tiền đề trình bày trong chương 2 và 3.

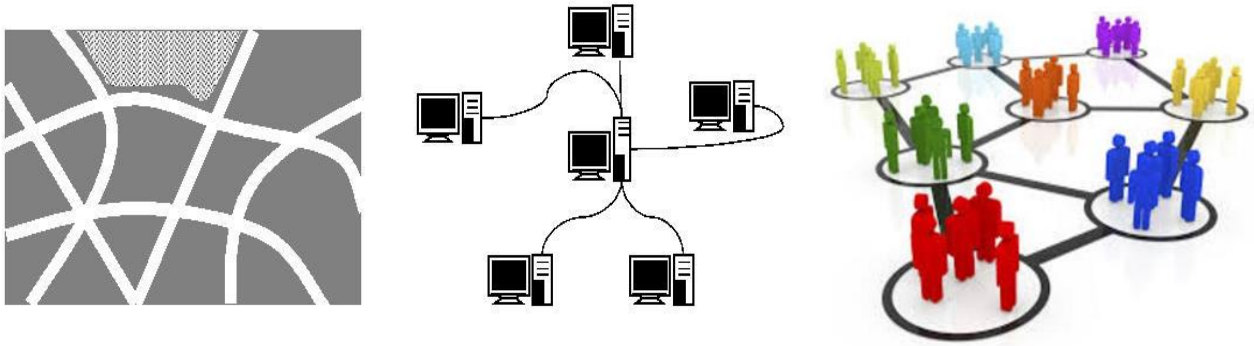
1.1. Một số khái niệm liên quan đến đồ thị

Lý thuyết đồ thị là một lĩnh vực nghiên cứu đã có từ lâu và có nhiều những ứng dụng hiện đại. Những tư tưởng cơ bản của lý thuyết đồ thị được đề xuất vào những năm đầu của thế kỷ XVIII bởi nhà toán học người Thụy Sỹ - Leonhard Euler

1.1.1. Định nghĩa đồ thị [1]

Đồ thị là một cấu trúc rời rạc bao gồm các đỉnh và các cạnh nối giữa các đỉnh đó. Người ta thường ký hiệu đồ thị $G = (V, E)$, V là tập các đỉnh (*Verterx*), E là tập ác cạnh (*Edge*). Có thể coi E là tập các cặp (u, v) với u và v là hai đỉnh của V.

Một số hình ảnh về đồ thị:



Sơ đồ mạng giao thông | Sơ đồ mạng Internet | Sơ đồ mạng xã hội

Hình 1.1: Ví dụ về mô hình đồ thị [1]

1.1.2. Các loại đồ thị

Có thể phân loại đồ thị ở đặc tính và số lượng của tập các cạnh E: Cho đồ thị $G = (V, E)$. Định nghĩa một cách hình thức.

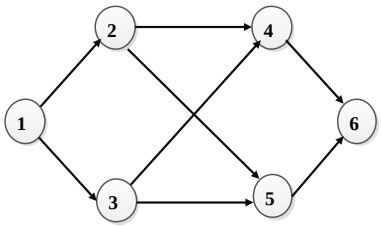
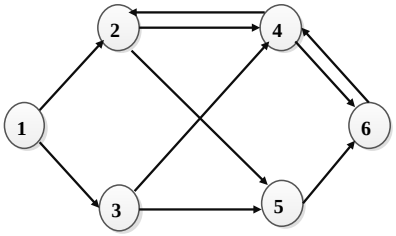
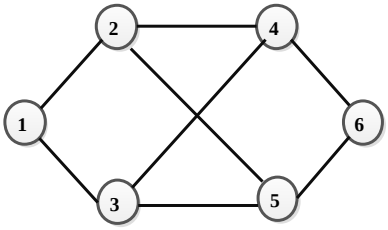
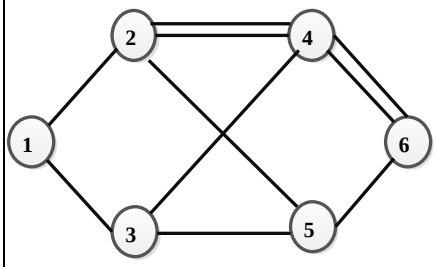
1. G được gọi là đơn đồ thị nếu giữa hai đỉnh u, v của V có nhiều nhất là 1

cạnh trong *E* nối từ *u* tới *v*.

2. *G* được gọi là đa đồ thị nếu giữa hai đỉnh *u, v* của *V* có thể có nhiều hơn 1 cạnh trong *E* nối từ *u* tới *v* (Hiển nhiên đơn đồ thị cũng là đa đồ thị).

3. *G* được gọi là đồ thị vô hướng nếu các cạnh trong *E* là không định hướng, tức là cạnh nối hai đỉnh *u, v* bất kỳ cũng là cạnh nối hai đỉnh *v, u.* Hay nói cách khác, tập *E* gồm các cặp *(u, v)* không tính thứ tự (*u, v) và (v, u).*

4. *G* được gọi là đồ thị có hướng nếu các cạnh trong *E* là có định hướng, có thể có cạnh nối từ đỉnh *u* tới đỉnh *v* nhưng chưa chắc đã có cạnh nối từ đỉnh *v* tới đỉnh *u*. Nói cách khác tập *E* gồm các cặp *(u, v)* có tính thứ tự: *(u, v) ≠ (v, u)*. Trong đồ thị có hướng, các cạnh được gọi là các cung. Đồ thị vô hướng cũng có thể coi là đồ thị có hướng nếu như ta coi cạnh nối hai đỉnh *u, v* bất kỳ tương đương với hai cung *(u, v) và (v, u).*

Đồ thị	**Đơn đồ thị**	**Đa đồ thị**
Có hướng		
Vô hướng		

Hình 1.2: Phân loại về đồ thị [1]

Một số dạng đồ thị đơn vô hướng đặc biệt:

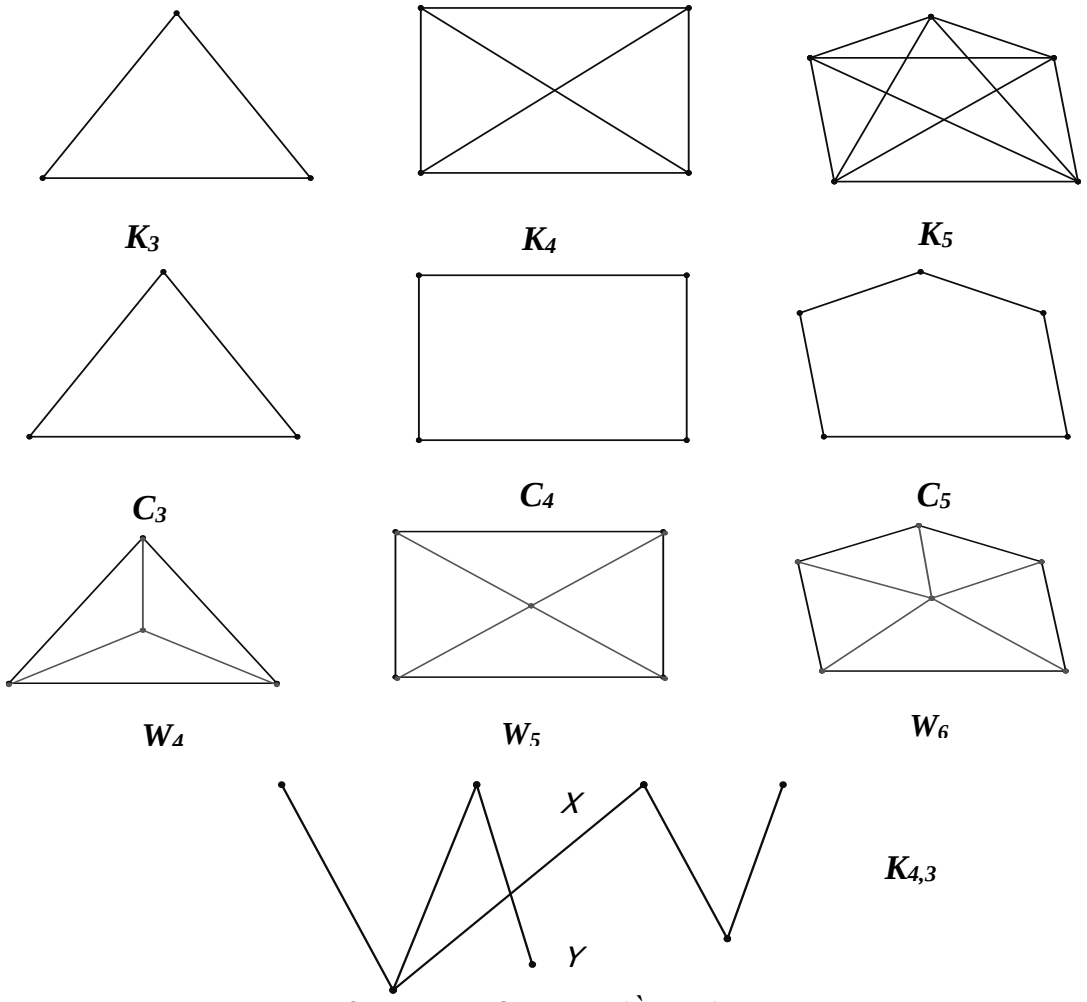
Đồ thị đầy đủ Kn *(compelte graph):* Là đơn đồ thị vô hướng mà giữa hai đỉnh bất kì của nó luôn tồn tại cạnh nối.

Đồ thị vòng Cn *(cycle graph):* Là đơn đồ thị vô hướng *G* = (*V, E*) với tập đỉnh *V* + {1, 2, 3,…, n} và tập cạnh *E* = {(1, 2); (2, 3); ….; (n – 1, n); (n, 1)}.

Đồ thị bánh xe Wn *(wheel graph):* là đơn đồ thị vô hướng thu được từ đồ

thị *Cn-1* bằng cách thêm một đỉnh *n* nối với *n-1* đỉnh của đồ thị *Cn-1*.

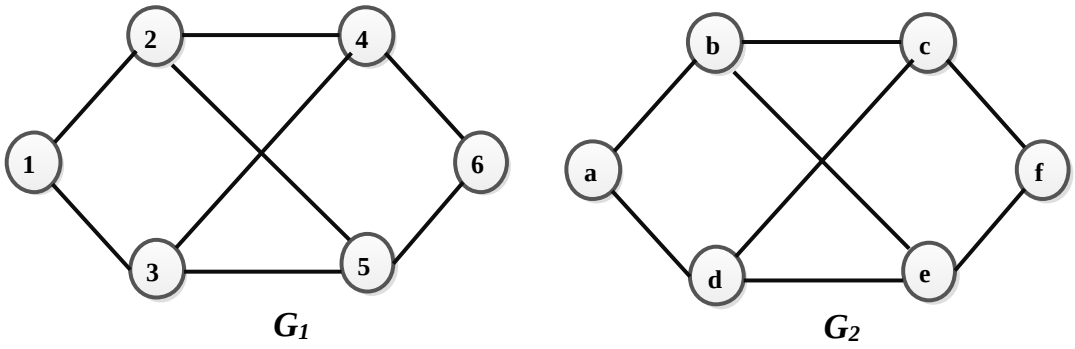
Đồ thị hai phía Km, n *(bipartite graph):* là đồ thị có tập đỉnh phân hoạch thành hai tập con không giao nhau $V=X \cup Y$ sao cho mọi cạnh nối một đỉnh thuộc *X* với một đỉnh thuộc *Y*.



Hình 1.3: Các dạng đồ thị đặc biệt [1]

1.1.3. Các khái niệm liên quan

Cho đồ thị $G = (V, E)$: trong đó có các tập đỉnh $V = \{1, 2, 3, ..., n\}$ và các tập cạnh $E = \{e_1, e_2, ..., e_n\}$. là một cấu trúc rời rạc, tức là các tập *V* và *E* hoặc là tập hữu hạn, hoặc là tập đếm được, có nghĩa là ta có thể đánh số thứ tự 1, 2, 3... cho các phần tử của tập *V* và *E*. Hơn nữa, đứng trên phương diện người lập trình cho máy tính thì ta chỉ quan tâm đến các đồ thị hữu hạn (*V* và *E* là tập hữu hạn), chính vì vậy nếu không chú thích thì khi nói tới đồ thị, ta hiểu rằng đó là đồ thị hữu hạn.



G_1 G_2

Hình 1.4: Các khái niệm liên quan đến đồ thị [1]

Cạnh (edge)

Nếu *(u, v)* là một cặp đỉnh thuộc *E* thì nói có một cạnh nối *u* và *v*. Khi đó *v* được gọi là kề của *u*.

Bậc của đỉnh

Gọi bậc của đỉnh trong đồ thị vô hướng là số cạnh liên thuộc với chính đỉnh đó và được kí hiệu là *deg(v)*.

Bán bậc của đỉnh

Bậc ra (vào) của đỉnh trong đồ thị có hướng là số cạnh của đồ thị đi ra (vào) đỉnh đó và kí hiệu là $deg^+(v)$ *hay* $deg^-(v)$. Ví dụ trong hình 1.4 đỉnh *2* của G_1 có bán bậc vào là 1: *hay* $deg^{--}(2)=1$ và bán bậc ra là 2: $deg^{2--}(2) = 2$.

Đường đi (path)

Một đường đi từ đỉnh *u* đến đỉnh *v* trên đồ thị *G* là một dãy đỉnh từ u_1, u_2,..., u_i. *Trong đó v có c*ác cạnh $(u, u_1), (u_1, u_2), ..., (u_i, v) \in E$, *và i* là số lượng cung trên đường đi được gọi là độ dài của đường đi.

Đường đi đơn

Một đường đi đơn trên đồ thị là một đường đi mà trên đó không có cạnh nào lặp lại.

Chu trình (cycle)

Một chu trình trên đồ thị *G* là một đường đi đơn có đỉnh đầu và đỉnh cuối trùng nhau.

Ví dụ trong hình 1.2 (Đơn đồ thị vô hướng ta có):

- Đường đi*: a →b→c→f→e→b→c*
- Đường đi đơn: *a→b→c→f→e→b*

- Chu trình: *b→c→f→e→b.*

Hai đỉnh liên thông

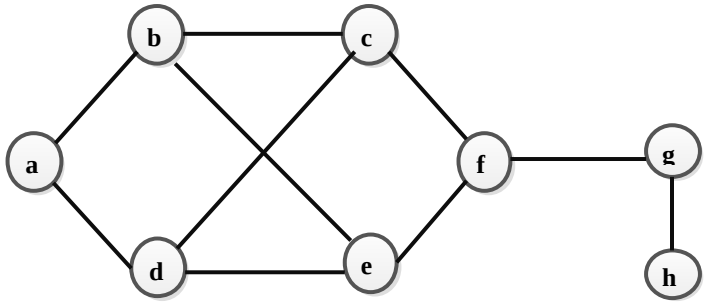
Đỉnh *p* và *q* được gọi là liên thông với nhau trên đồ thị *G* nếu có một đường đi từ *p* đến *q* trên đồ thị đó.

Đồ thị liên thông

Một đồ thị được gọi là liên thông nếu mọi cặp đỉnh của đồ thị đều liên thông.

Thành phần liên thông

Đồ thị *G* không liên thông sẽ phân rã thành một số đồ thị con hữu hạn liên thông không có đỉnh chung. Các đồ thị con này được gọi là các thành phần liên thông của đồ thị.



G

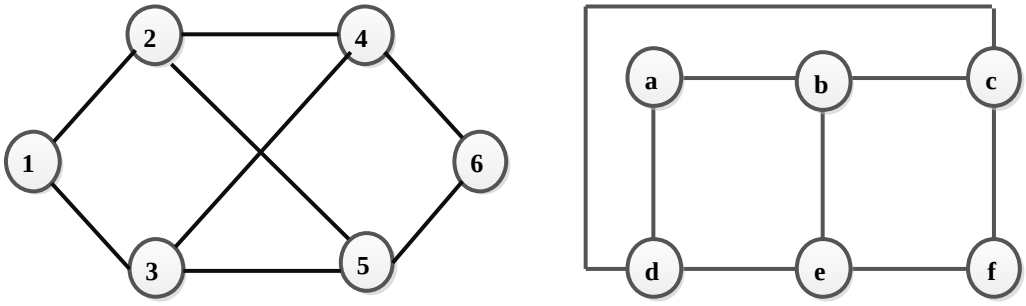
Hình 1.5. Đỉnh rẽ nhánh và bắc cầu [1]

Đỉnh rẽ nhánh

Đỉnh *u* được gọi là đỉnh rẽ nhánh của đồ thị *G* nếu việc loại bỏ đỉnh đó cùng các cạnh liên thuộc với nó làm tăng số thành phần liên thông của đồ thị.

Cầu

Cạnh *e* được gọi là cầu của đồ thị *G* nếu việc loại bỏ cạnh đó làm tăng số thành phần liên thông của đồ thị**.**



G₁ ***G₂***

Hình 1.6. Đồ thị con và đồ thị đẳng cấu [1]

Đồ thị con

Đồ thị $H= (W, F)$ được gọi là đồ thị con của đồ thị $G = (V, E)$ nếu $W \subseteq V$ và $F \subseteq E$.

Đồ thị đẳng cấu

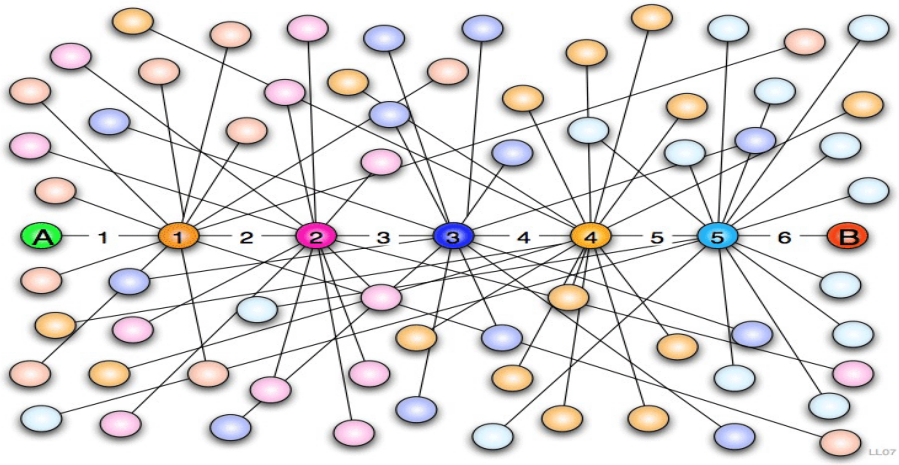
Hai đồ thị $G_1= (V_1, E_1)$ và $G_2= (V_2, E_2)$ được gọi là đẳng cấu nếu tồn tại một song ánh $f: E_1 \rightarrow E_2$ sao cho $(u, v) \rightarrow E_1$ khi và chỉ khi $(f(u), f(v)) \rightarrow E_2$.

1.2. Một số khái niệm liên quan về mạng xã hội

Mạng xã hội xuất hiện trong nhiều lĩnh vực như: Xã hội học, Công nghệ thông tin (khai phá dữ liệu), khoa học hành vi, toán học, thống kê và nhiều lĩnh vực khác. Mạng xã hội (Social network sites), mạng xã hội trên Internet, mạng xã hội trực tuyến, hay còn gọi là mạng xã hội ảo, là một khái niệm mới được hình thành trong thập niên cuối thế kỷ XX, bắt đầu bằng sự ra đời của Classmates.com (1995), SixDegrees (1997), kế đến là sự nở rộ của một loạt các trang mạng khác như Friendster (2002), Facebook (2004), Twitter (2006) và tại Việt Nam Zing me (2009) [2]… với sự phát triển nhanh chóng của các hình thức xã hội ảo này nên mạng xã hội được định nghĩa rất khác nhau tùy theo hướng tiếp cận.

Một cách chung nhất mạng xã hội là tập hợp các cá nhân với các mối quan hệ về một hay nhiều mặt gắn kết với nhau. Mạng xã hội là một bản đồ của tất cả các mối quan hệ liên quan giữa tất cả các nút đang được nghiên cứu, mạng cũng có thể được sử dụng để đo vốn xã hội – giá trị mà các cá nhân có từ mạng xã hội, được hiển thị trong một sơ đồ mạng xã hội, nơi mà các nút là các điểm và quan hệ là các đường.

Về mặt toán học, mạng xã hội có thể xem như một hệ thống các điểm (node) gắn với nhau thành một mạng gồm các liên kết (hoặc các cung). Theo hướng tiếp cận này mạng xã hội được xem như mạng phức hợp, hay nói cách khác là một tập các hệ thống được tạo bởi các yếu tố đồng nhất hoặc không đồng nhất kết nối với nhau thông qua sự tương tác khác nhau giữa các yếu tố này và được trải ra trên diện rộng. Mạng phức hợp có 2 thuộc tính quan trọng là “hiệu ứng thế giới nhỏ” (small – world effect) và “đặc trưng co giãn tự do” (Scale – free feature).



Hình 1.7: Ma trận mạng xã hội

(https://upload.wikimedia.org/wikipedia/commons/9/94/Six_degrees_of_separation.png)

1.2.1. Phân tích cấu trúc mạng xã hội

Một mạng xã hội là một bản đồ của các mối quan hệ nhất định chẳng hạn mối qua hệ giữa các nút như tính liên kết giữa các nút đang được nghiên cứu. Các mối quan hệ mà cá nhân như là các nút các kết nối là những quan hệ xã hội của cá nhân đó. Mạng lưới này cũng có thể được sử dụng để đo lường vốn xã hội những giá trị mà một cá nhân nhận được từ các mạng xã hội. Những khái niệm về phân tích mạng xã hội thường được hiển thị trong một sơ đồ mạng xã hội, nơi mà các nút là các điểm và các mối quan hệ là các dòng.

Có nhiều kiểu để phân tích mạng xã hội: phân tích dựa trên liên kết và câu trúc; phân tích dựa trên nội dung; phân tích kết hợp.

Phân tích mạng xã hội (liên quan đến lý thuyết mạng) đã nổi lên như là một kỹ thuật quan trọng trong xã hội học hiện đại. Dựa trên việc phân tích và mô phỏng mà người ta đã áp dụng trong nhiều lĩnh vực quan trọng như trong nhân văn học, sinh học, nghiên cứu truyền thông, kinh tế, địa lý, khoa học thông tin, nghiên cứu tổ chức, tâm lý xã hội và xã hội học. Người ta đã dùng ý tưởng "mạng xã hội" lỏng lẻo trong hơn một thế kỷ để bao hàm bộ phức tạp của các mối quan hệ giữa các thành viên của hệ thống xã hội ở tất cả các quy mô, từ cá nhân đến quốc tế.

Năm 1954, J. A. Barnes bắt đầu sử dụng thuật ngữ có hệ thống để biểu thị mô hình quan hệ, bao gồm các khái niệm truyền thống được sử dụng bởi công

chúng và các nhà khoa học xã hội: ghép các nhóm với nhau (ví dụ, các bộ lạc, gia đình) và loại xã hội (ví dụ, giới tính, dân tộc). Các học giả như S.D. Berkowitz,...

Phân tích mạng xã hội hiện nay đã chuyển từ cách tiếp cận đánh giá những mô hình bằng phương pháp riêng trong việc dùng mô phỏng qua phần mềm phân tích mạng xã hội. Như phân tích cấu trúc giữa các mối quan hệ cá nhân từ những hành vi thái độ, phân biệt giữa các đối tượng, các nhóm …Bằng việc phân tích trên các nhà phân tích dự kiến sẽ có đầy đủ thông tin về những người đang ở trong mạng, tất cả những người tham gia kể cả những thành viên liên kết nhóm.

Một số xu hướng phân biệt trong phân tích mạng xã hội:

Việc tiếp cận các mối quan hệ là sự tiếp cận liên kết trong cộng đồng và không bị giới hạn cũng như việc liên kết giữa các trang web. Việc phân tích là tập trung vào cấu trúc giữa các mối quan hệ và cho thấy cấu trúc thành phần mối quan hệ ảnh hưởng ở mức độ nào đó.

Hình dạng của mạng xã hội giúp xác định sự hiệu quả của mạng lưới các cá nhân của mình. Trong giới hạn nào đó, mạng lưới chặt chẽ hơn nhỏ hơn có thể hữu ích cho các thành viên trong nhóm. Việc cởi mở liên kết với thành viên ngoài nhóm nảy sinh những mối quan hệ kết nối lỏng lẻo. Do vậy cần có những ý tưởng thiết lập nên mối quan hệ theo mạng lưới liên kết với nhiều mối quan hệ ngoài mà không ảnh hưởng đến các nhóm liên kết giữa các thành viên nhóm. Một nhóm cá nhân có liên hệ với thế giới xã hội khác có thể có quyền truy cập vào một phạm vi rộng của thông tin để kết nối tới nhiều mạng lấy thông tin bằng cách bắc cầu mà không cần trực tiếp liên kết.

Sức mạnh của phân tích mạng xã hội xuất phát từ sự khác biệt của các nghiên cứu khoa học xã hội truyền thống, đã giải thích được nhiều hiện tượng trong thế giới thực.

Phân tích mạng xã hội đã được sử dụng trong dịch tễ học giúp hiểu mô hình của con người liên lạc hỗ trợ hoặc ức chế sự lây lan của các bệnh như HIV trong dân chúng như thế nào. Sự tiến hóa của mạng xã hội đôi khi có thể được

mô hình bằng cách sử dụng dựa trên những mô hình thực thể, sự tương tác giữa quy tắc giao tiếp, thông tin lan rộng và cơ cấu xã hội.

Một số nhà nghiên cứu đã đề xuất rằng các mạng xã hội của con người có thể có một cơ sở di truyền. Sử dụng một mẫu của cặp song sinh trong độ tuổi vị thành niên từ các quốc gia khác nhau cho thấy rằng ở một mức độ nào đó xác suất giữa hai người là bạn của nhau. Việc phân tích các mô hình cặp song sinh như một mạng lưới di truyền trong đó các cặp là những nút đa dạng mô phỏng các tính năng khác của con người.

Số liệu thống kê trong phân tích mạng xã hội [9]

a. Thuật ngữ:

Vai trò trung tâm

Mức độ mà một nút nằm giữa các nút khác trong mạng. Biện pháp này để tính toán kết nối giữa các nút lân cận, đưa ra một giá trị cao hơn cho các nút cụm. Các biện pháp phản ánh số người sử dụng có một người kết nối gián tiếp thông qua liên kết trực tiếp.

Liên kết

Một cạnh được liên kết với nhau nếu xóa cạnh liên kết giữa chúng sẽ tạo ra hai điểm cuối nằm trong các thành phần khác nhau của đồ thị.

Tính trung tâm

Là kết cấu giữ vai trò đại diện bao trùm quan sát các vị trí xung quanh.

Tính tập trung

Sự khác biệt giữa số lượng các liên kết cho mỗi nút là sự khác biệt khi chia cho tối đa. Một mạng lưới tập trung sẽ có nhiều người trong số các liên kết phân tán xung quanh một hoặc một vài nút, trong khi một mạng lưới phân mức là một hay các biến thể nhỏ giữa số lượng các liên kết mỗi nút sở hữu.

Tính chính xác

Mức độ một cá nhân nằm gần tất cả các cá nhân khác trong một mạng lưới (trực tiếp hoặc gián tiếp). Nó phản ánh khả năng truy cập thông tin “cung mức tin” của các thành viên mạng lưới. Vì vậy lân cận là nghịch đảo của tổng số

khoảng cách ngắn nhất giữa mỗi cá nhân và mọi người khác trong mạng. Đường đi ngắn nhất cũng có thể được gọi là “khoảng cách trắc địa cực tiểu”.

Cụm hệ số

Một thước đo khả năng có kết giao của một nút liên kết phụ thuộc bản thân nó. Một yếu tố kết cụm cao chỉ ra một liên kết nhóm lớn.

Gắn kết

Mức độ mà các thành viên kết nối trực tiếp với nhau bằng bằng các kết cấu trái chiều. Nhóm được xác định nếu mỗi cá nhân trực tiếp gắn liền với mỗi cá nhân khác, “vòng kết nối xã hội” chặt chẽ hơn khi tiếp xúc trực tiếp các khối cấu trúc được gắn kết chính xác hoặc các khối cấu trúc gắn kết không chính xác.

Bậc

Là số lượng đếm các quan hệ với các thành viên khác trong mạng. Trong lý thuyết đồ thị gọi là mật độ phân tử (bậc cá thể).

Mật độ

Mức độ quan hệ mối quan hệ hiểu biết lẫn nhau trên tỷ lệ của mối quan hệ giữa các cá nhân. Mạng hoặc mức độ phân tử là tỷ lệ quan hệ trong một mạng lưới tương đối so với tổng thể (mạng lưới thưa thớt so với mạng lưới dày đặc).

Dòng vai trò trung tâm

Mức độ mà một nút đóng góp vào tổng lưu lượng tối đa giữa tất cả các cặp nút trung tâm.

Một thước đo về tầm quan trọng của một nút trong mạng nó gán điểm số tương đối so với tất cả các nút trong mạng dựa trên nguyên tắc các kết nối với các nút có một điểm số cao đóng góp nhiều hơn điểm số của các nút trong nó.

Kết nối vùng lân cận

Một cạnh là một kết nối lân cận nếu các điểm cuối chia sẻ không có lân cận chung. Không giống như một lân cận một liên kết lân cận được chứa trong một chu kỳ.

Chiều dài đường

Khoảng cách giữa các cặp nút trong mạng là mức trung bình của các khoảng cách giữa các cặp nút.

Uy tín

Trong một đồ thị uy tín là thuật ngữ dùng để mô tả vai trò trung tâm của một nút. “Mức độ uy tín”, “tiệm cận uy tín”, và “tình trạng uy tín” là tất cả các biện pháp của uy tín.

Thông tin

Bằng liên kết mạng thông tin cá nhân được cập nhật và các liên kết đều được biết sớm.

Tiệm cận

Mức độ tiếp cận bất kỳ thành viên trong một mạng có thể tiếp cận được các thành viên khác trong cùng mạng.

Sự gắn kết cấu

Số lượng tối thiểu của các thành viên nếu bị xóa khỏi nhóm sẽ ngắt kết nối nhóm.

Cấu trúc tương đối

Đề cập đến mức độ mà các nút có một tập hợp của các mối liên kết với các nút khác trong hệ thống. Các nút không cần phải có bất kỳ mối quan hệ với nhau để có cấu trúc tương đương.

Kết cấu lỗ

Kết cấu lỗ tĩnh có thể được lấp đầy chiến lược bằng cách kết nối một hoặc nhiều liên kết để liên kết với nhau ở điểm khác. Liên kết với ý tưởng trong xã hội: Nếu bạn liên kết với những người không có hai liên kết, bạn có thể kiểm soát thông tin liên lạc của họ.

Trong biểu đồ phân tích lý thuyết và mạng, không có các biện pháp khác nhau của vị trí trung tâm của một đỉnh trong cùng một đồ thị. Xác định tầm quan trọng tương đối của một đỉnh trong đồ thị (ví dụ, làm thế nào một người quan trọng là trong một mạng xã hội, hoặc trong lý thuyết về cú pháp không gian, làm thế nào một căn phòng quan trọng nằm trong một tòa nhà, hoặc làm thế nào được sử dụng một con đường tốt nằm trong một mạng lưới đô thị).

b. Vị trí trung tâm

Việc đầu tiên và đơn giản nhất là mức độ trung tâm. Vị trí trung tâm được

định nghĩa là số lượng các liên kết vào một nút (ví dụ: một nút nếu có số lượng các mối quan hệ). Thông qua nút thì các nguy cơ mất an toàn giữa các liên kết cũng không cao trong hệ thống mạng (chẳng hạn như bị lây virus, hoặc thông tin bị rò rỉ...). Nếu trong hệ thống mạng được quan tâm chặt chẽ (có nghĩa là mối quan hệ có hướng), ta thường xác định hai biện pháp riêng biệt của vị trí trung tâm, cụ thể là ở mức độ vào và mức độ ra. Mức độ vào là việc đếm số lượng các mối quan hệ trực tiếp tới nút và mức độ ra là số quan hệ mà nút hướng đến những người khác. Đối với mối quan hệ tích cực như tình bạn hoặc liên kết nhóm tư vấn, ta thường giải thích mức độ vào như một hình thức phổ biến và mức độ ra như kết thành đám.

Cho đồ $G := (V, E)$ bằng n đỉnh có mức độ trung bình $C_D(v)$ của đỉnh v.

$$C_D(v) = \frac{\deg(v)}{n-1} \tag{1.1}$$

Tính mức độ trung bình cho tất cả các giao điểm V trong một đồ thị $\Theta(V^2)$. Một ma trận kề của đồ thị cho các cạnh E trong một đồ thị có $\Theta(E)$ ma trận.

Khi đó tính mức độ trung bình của đồ thị có v^* là nút trung tâm của G, cho $X:=(Y;Z)$ là n nút kết nối đồ thị nhằm tối đa số lượng (với y^* là nút ở mức trung tâm trong X).

$$H = \sum_{j=1}^{|Y|} C_D(y^*) - C_D(y_j) \tag{1.2}$$

Khi đó giá trị trung bình của đồ thị G được tính như sau:

$$C_D(G) = \frac{\sum_{i=1}^{|V|} [C_D(v^*) - C_D(v_i)]}{H} \tag{1.3}$$

H có giá trị lớn nhất khi đồ thị X có số nút được kết nối với tất cả các nút và các nút được kết nối tới một vị trí trung tâm nút (một ánh xạ của đồ thị) trong cây.

$$H = (n-1)(1-\frac{1}{n-1}) = n-2 \tag{1.4}$$

Do đó vị trí trung tâm của G bị giảm

$$C_D(G) = \frac{\sum_{i=1}^{|V|} [C_D(v^*) - C_D(v_i)]}{n-2} \tag{1.5}$$

Khi xác định một vị trí trung tâm của một đỉnh bằng đồ thị các đỉnh diễn ra được liên kết với nhiều hướng khác nhau.

Cho một đồ thị *G := (V,E)* bằng tập *n* đỉnh trong đó $C_B(v)$ của đỉnh *v* được tính như sau:

- Đối với mỗi cặp của đỉnh (*s, t*) tính được đường đi ngắn nhất.

- Đối với mối cặp cuả đỉnh (*s, t*) xác định các đường đi ngắn nhất thông qua đỉnh *v*.

- Tổng của tất cả các đỉnh (*s, t*) hoặc ngắn hơn.

$$C_b(v) = \sum_{s \neq v \neq t \in V} \frac{\delta_{st}(v)}{\delta_{st}} \qquad (1.6)$$

Khi đó δ_{st} là số thứ tự từ *s* tới *t* và $\delta_{st}(v)$ là số thứ tự từ *s* tới *t* đi qua *a* đỉnh *v*. Do đó (*n-1*)(*n-2*) trong đồ thị có hướng và (*n-1*)(*n-2*)/2 cho đồ thị vô hướng. Ví dụ: trong đồ thị vô hướng có đỉnh ở giữa, vị trí trung tâm được tính (*n-1*)(*n-2*)/2 không chứa 0 và có tỷ lệ đường đi ngắn nhất.

Tính vị trí trung tâm và điểm gần nhất của tất cả các đỉnh liên quan đến việc tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trên đồ thị đó. Trong khi $\Theta(V^3)$ là thời gian mà thuật toán Floyd-Warshall để thay đổi và tìm tất cả các đường đi ngắn nhất giữa hai nút trên đồ thị. Thuật toán của Johnson cho kết quả tính toán $O(V^2 logV+VE)$ thời gian. Trong liên kết đồ thị vị trí trung tâm được tính toán với $O(VE)$ thời gian.

Trong đồ thị vô hướng và kết nối với nhiều cạnh, phép tính vị trí trung tâm và liền kề của tất cả các đỉnh trong đồ thị khi đó được cụ thể với các mạng đồ thị trong trường hợp này ta sử dụng thuật toán Brandes để tính điểm trung tâm và mỗi đường đi ngắn nhất được tính hai lần.

c. Vị trí gần trung tâm

Trong cấu trúc liên kết và các khu vực liên quan đến toán học sự gần gũi là một trong những khái niệm cơ bản trong một không gian. Bằng trực giác ta nói hai tập hợp gần nếu chúng kề nhau. Khái niệm này có thể được định nghĩa một cách tự nhiên trong một không gian diện tích với một khái niệm về khoảng cách

giữa các thành phần của không gian được xác định, nhưng nó có thể được tổng quát cho không gian hình học nơi ta không có cách nào cụ thể để đo khoảng cách.

Trong biểu đồ lý thuyết lân cận là một biện pháp trung tâm của một đỉnh trong một đồ thị. Việc tìm kiếm lân cận được ưa thích trong phân tích mạng có nghĩa là chiều dài đường đi ngắn nhất vì mang lại giá trị cao hơn với các đỉnh trung tâm và do đó thường được kết hợp với các biện pháp khác nhau như mức độ.

Trong lý thuyết mạng gần gũi là một biện pháp tinh vi của vị trí trung tâm. Nó được định nghĩa là khoảng cách đo trung bình (tức là đường đi ngắn nhất) giữa một đỉnh *v* và tất cả các đỉnh khác có thể truy cập từ nó:

$$\frac{\sum_{t \in V \setminus v} d_G(v,t)}{n-1} \qquad (1.7)$$

Từ $n \geq 2$ là kích thước của mạng được kết nối với cấu tạo *V* và được truy cập từ *v*. Sự gần gũi có thể được coi là một biện pháp lấy thông tin trong thời gian bao lâu để từ một đỉnh có thể truy cập cho đỉnh khác trong mạng .

Một số định nghĩa lân cận là nghịch đảo của số lượng, nhưng cách thức thông tin truyền đạt là như nhau. Sự gần gũi cho một đỉnh là đối ứng của tổng khoảng cách đo đến tất cả các đỉnh khác của *V*.

$$C_C(v) = \frac{1}{\sum_{t \in V \setminus v} d_G(v,t)} \qquad (1.8)$$

d. Giá trị đặc trưng trung tâm

Là một biện pháp quan trọng của một nút trong một mạng chỉ định tương đối cho tất cả các nút trong mạng dựa trên nguyên lý kết nối để ghi nhiều hơn đến các điểm nút.

Sử dụng ma trận kề để tìm vị trí trung tâm của vector đặc trưng.

Để x_i biểu thị số điểm nút thứ i là ma trận kề của mạng A_{ij}. Do đó $A_{ij} = 1$ nếu nút thứ i là bên cạnh nút j và $A_{ij} = 0$. Nói chung các mục trong A có thể đại diện cho kết nối mạnh như trong một ma trận số thực ngẫu nhiên.

Đối với các nút i^{th} cho phép vị trí điểm trung tâm tỉ lệ thuận với tổng số điểm của tất cả các nút được kết nối với nó.

Do đó:

$$x_i = \frac{1}{\lambda} \sum_{j \in M(i)} x_j = \frac{1}{\lambda} \sum_{j=1}^{N} A_{ij} xj \qquad (1.9)$$

Khi $M(i)$ là tập hợp các nút i^{th} được kết nối N là tổng số nút và λ là hằng số vector này được viết như sau:

$$x = \frac{1}{\lambda} Ax \qquad (1.10)$$

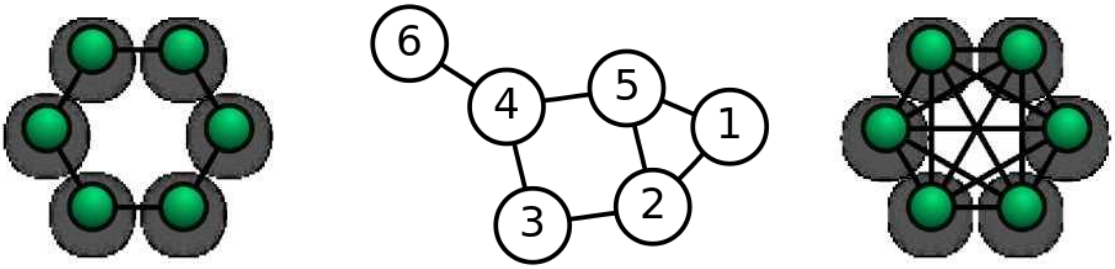
Hoặc phương trình $Ax = \lambda x$.

Như vậy: có nhiều giá trị khác nhau đối với mỗi giải pháp khác nhau trong vertor tồn tại, tuy nhiên các đặc điểm trung tâm của các nút trong mạng. Việc sử dụng nhiều tần suất lặp trong các thuật toán có thể được sử dụng để tìm giá trị vượt trội của vector đặc trưng.

1.2.2. Biểu diễn độ phân rã về mạng xã hội trên đồ thị

Sự gắn kết hay kết cấu gắn kết là xã hội học và khái niệm về lý thuyết đồ thị đo lường của việc gắn kết cho nhóm tối đa xã hội hoặc ranh giới biểu diễn bằng đồ họa bởi các yếu tố liên quan không bị ngắt kết nối, chỉ loại bỏ số lượng ở tối thiểu một số nút nhất định. Các giải pháp cho sự gắn kết được tìm thấy bởi cắt các đỉnh trong định lý của *Menger*. Các ranh giới của cấu trúc là một trường hợp đặc biệt của sự gắn kết. Cũng rất hữu ích khi biết đồ thị *k-gắn kết* (hoặc *k-thành phần*) luôn luôn là một đồ thị con của một *k-core*, mặc dù một *k-gắn kết* không phải lúc nào là *k-core*. Một *k-core* chỉ đơn giản là một đồ thị con trong đó tất cả các nút có ít nhất *k* lân cận nhưng nó không cần được kết nối.

Ví dụ:



Vòng 6 nút trong đồ thị có khả năng gắn kết nối 2 hoặc trên mức 2 cũng có thể loại bỏ hai nút là cần thiết để ngắt kết nối chúng.

Phần 6-nút (1 kết nối) có một nút nhúng vào 2 thành phần, các nút 1-5.

Một nhóm 6-nút là một gắn kết 5 thành phần, có cấu trúc 5.

Hình 1.8: Biểu diễn độ phân rã bằng đồ thị [9].

a. Mô tả trong lý thuyết đồ thị

Trong toán học và khoa học máy tính lý thuyết đồ thị là nghiên cứu của đồ thị: Các cấu trúc toán học được sử dụng để mô hình hóa các mối quan hệ cặp giữa các đối tượng từ một tập xác định. Một "đồ thị" đề cập đến một bộ tập các đỉnh hoặc "nút" và một tập các cạnh nối các cặp đỉnh. Một đồ thị có thể vô hướng, có nghĩa không có sự phân biệt giữa hai đỉnh kết hợp với mỗi cạnh, hoặc cạnh của nó có thể được kết nối từ một đỉnh khác; Đồ thị trong toán học được định nghĩa chi tiết hơn và có các biến thể khác trong các loại biểu đồ thường. Các đồ thị nghiên cứu trong lý thuyết đồ thị không nên nhầm lẫn với "các chức năng của đồ thị" và các loại khác đồ thị.

Đồ thị là một trong các đối tượng chính được nghiên cứu trong toán học rời rạc. Ta có thể tham khảo bảng thuật ngữ lý thuyết đồ thị cơ bản định nghĩa trong lý thuyết đồ thị.

Vẽ đồ thị

Đồ thị được biểu diễn đồ họa bằng cách vẽ một dấu chấm cho mỗi đỉnh và vẽ một cung giữa hai đỉnh nếu chúng được kết nối bởi một cạnh. Nếu biểu diễn đồ thị có hướng chỉ cần vẽ bằng một mũi tên.

Vẽ cấu trúc đồ thị không nên nhầm với các đồ thị chính cấu trúc đồ thị đó, có rất nhiều cách để vẽ cấu trúc vẽ đồ thị. Tất cả những vấn đề như đỉnh được kết nối với các đỉnh khác bằng cạnh và cách bố trí cạnh kề không chính xác.

Cấu trúc dữ liệu trong đồ thị

Có những cách khác nhau để lưu trữ các đồ thị trong một hệ thống máy tính. Cấu trúc dữ liệu được sử dụng phụ thuộc vào cấu trúc của đồ thị và thuật toán được sử dụng cho các thao tác với đồ thị. Về lý thuyết có thể phân biệt giữa cấu trúc danh sách và ma trận, nhưng trong các ứng dụng cụ thể có cấu trúc tốt nhất thường là sự kết hợp của cả hai. Cấu trúc danh sách thường hay dùng cho đồ thị thưa với yêu cầu bộ nhớ nhỏ hơn. Mặt khác cấu trúc ma trận cung mức truy cập nhanh hơn cho một số ứng dụng nhưng có thể tiêu thụ một lượng lớn bộ nhớ.

Cấu trúc danh sách

Danh sách tỷ lệ

Các cạnh được đại diện bởi một mảng chứa cặp đỉnh (có cạnh nối) và có thể cả trọng lượng và dữ liệu khác, đỉnh là kết nối bởi một cạnh được liên kết liền kề.

Danh sách kề

Cũng giống như danh sách tỷ lệ mỗi đỉnh có một danh sách các đỉnh kề. Điều này gây ra dư thừa trong một đồ thị vô hướng: ví dụ, nếu đỉnh A và B đang cận kề khi đó sẽ có một danh sách kề chứa B và trong đó danh sách của B chứa A. Do vậy điểm cận kề là để truy vấn nhanh hơn với chi phí không thêm gian lưu trữ.

Cấu trúc ma trận

Ma trận liên thuộc

Các đồ thị được biểu diễn bằng một ma trận kích thước bằng *IV* |(Số đỉnh) của *IEI* (số cạnh) có thông tin vào, ra [đỉnh, cạnh] chứa dữ liệu thiết bị đầu cuối của cạnh (đơn giản: 1 - sự cố, 0 - không phải sự cố).

Ma trận kề

Đây là giao diên *n* bởi *n* ma trận *A*, trong đó *n* là số đỉnh của đồ thị. Nếu có một cạnh từ một đỉnh *x* đến một đỉnh *y*, các yếu tố a_{zy} là 1 (là số cạnh *xy*), nếu

nó là *0*. Trong máy tính dễ dàng tìm thấy đồ thị con, và đảo ngược thành một đồ thị có hướng.

Phương trình ma trận hay nguyên lý ma trận hoặc tổng ma trận.

Điều này được định nghĩa là *D* - *A* trong đó *D* là ma trận có đường chéo chính (được gọi là "phương trình ma trận" của một đồ thị.)

Khoảng cách ma trận

A đối xứng với *n* của *n* bởi ma trận *D* khi mà d_{zy} là độ dài của đường đi ngắn nhất giữa *x* và *y*; nếu không có đường đi thì d_{zy} bằng vô cực. Nó có thể được bắt nguồn từ điểm *A*.

$$d_{x,y} = \min\{n \mid A^n[\mathrm{x,y}] \neq 0\} \qquad (1.11)$$

b. Lý thuyết mạng

Lý thuyết mạng là một lĩnh vực của khoa học máy tính và khoa học mạng cũng là một phần của lý thuyết đồ thị. Nó có ứng dụng trong nhiều lĩnh vực bao gồm vật lý hạt nhân, khoa học máy tính, sinh học, kinh tế, hoạt động nghiên cứu, và xã hội học. Mối quan tâm của lý thuyết mạng riêng với việc nghiên cứu các đồ thị như một đại diện của các quan hệ đối xứng và giữa các đối tượng có quan hệ rời rạc bất cân xứng. Các ứng dụng của lý thuyết mạng bao gồm mạng lưới hậu cần, World Wide Web, mạng lưới trao đổi chất, các mạng xã hội, mạng lưới tri thức luận, ...

c. Tối ưu hóa mạng

Vấn đề mạng liên quan đến việc tìm kiếm một cách tối ưu hóa tổ hợp. Các ví dụ bao gồm lưu lượng mạng, vấn đề đường đi ngắn nhất, vấn đề giao thông, vấn đề chuyển tải, vấn đề vị trí, vấn đề phân công, đóng gói vấn đề, vấn đề định tuyến, phân tích đường dẫn quan trọng và PERT (Program Evaluation & Review Technique).

Phân tích mạng xã hội

Phân tích mạng xã hội là bản đồ mối quan hệ giữa các cá nhân trong các mạng xã hội. Những người này thường có thể là các nhóm (bao gồm cả các nhóm cộng đồng và các khối có gắn kết), các tổ chức, các quốc gia, các trang web, hoặc

trích dẫn từ các ấn phẩm học thuật. Mạng lưới phân tích là phân tích lưu lượng liên kết lân cận, việc theo dõi thông qua giữa các nút mạng và cấu trúc có thể được thiết lập. Điều này có thể được sử dụng để phát hiện ra các mạng lưới con.

d. Luồng trên mạng

Trong lý thuyết đồ thị một luồng trên mạng là một đồ thị có hướng mà mỗi cạnh nhận được một đường đi. Đường đi của một cạnh không vượt quá khả năng của cạnh. Thông thường trong hoạt động nghiên cứu một đồ thị có hướng được gọi là một mạng lưới, các đỉnh được gọi là các nút và các cạnh được gọi là vòng cung. Một dòng chảy phải đáp ứng các hạn chế đó lượng dòng chảy vào một nút bằng với lượng dòng chảy ra khỏi nó, trong đó có dòng chảy ra nhiều hơn, hoặc có nhiều dòng chảy đến hơn. Một mạng lưới có thể được sử dụng để mô hình trong một hệ thống đường giao thông, chất lỏng trong ống, dòng điện trong một mạch điện hoặc bất cứ điều gì tương tự.

e. Định nghĩa

Cho $G(V,E)$ là đồ thị hữu hạn trong đó mỗi cạnh $(u,v) \in E$ là một giá trị thực không âm của $c(u,v)$. Nếu $(u, v) \notin E$, thì c$(u, v) = 0$ Có hai đỉnh: *a* của *S* và *a* của *t* một luồng là một hằng số thực: $f: VxV \rightarrow \omega$ bằng với ba thuộc tính cho tất cả các nút *v* và *u:*

Khả năng $f(u, v) < c(u, v)$. Các dòng chảy dọc theo một cạnh không thể vượt quá khả năng của mình.

Hạn chế:

Đối xứng: $f(u, v) = -f(u, v)$. Dòng chảy tới *u* từ *v* và ngược lại từ v tới u.

Bảo tồn: $\sum_{\omega \in V} f(v, \omega) = 0$ trừ khi $u = s$ hoặc $w = t$ dòng chảy đến một nút bằng 0.

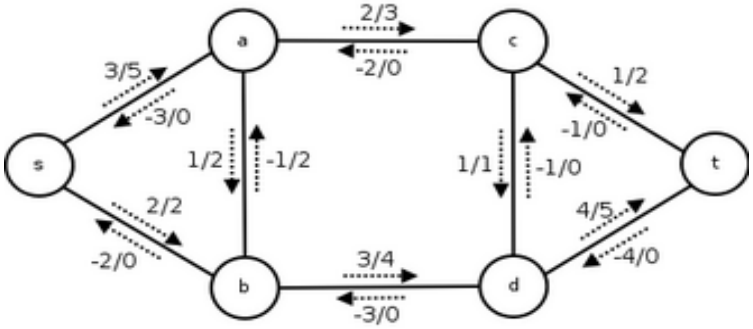
Nếu đồ thị đại diện cho một mạng vật lý và nếu có một dòng chảy thực tế cho ví dụ 4 đơn vị từ *u* đến *v* và là dẫy thực của 3 đơn vị từ *v* tới *u*, ta có $f(u,v)=1$ và $f(v,u) = -1$.

Có các cạnh dư là $C_f(u, v) = C(u, v) — f(u. v)$, một mạng còn dư được ký hiệu là $G_f(V, E_f)$ ta thấy rằng có một cạnh từ *u* đến *v* trong mạng còn lại mặc dù không có cạnh từ *u* đến *v* trong mạng ban đầu. Do dòng chảy theo hướng

ngược nhau bị hủy bỏ, giảm dòng chảy từ *u* đến *v* là tăng lưu lượng từ *u* đến *v*. Con đường trong mạng thông khi (u_1, u_2,...u_k) dư trong mạng, và $u_1 = s$, $u_k = t$, $c_f(u_i, u_{i+1})>0$. Một mạng có lưu lượng tối đa nếu và chỉ nếu không có con đường làm tăng trong các mạng còn lại.

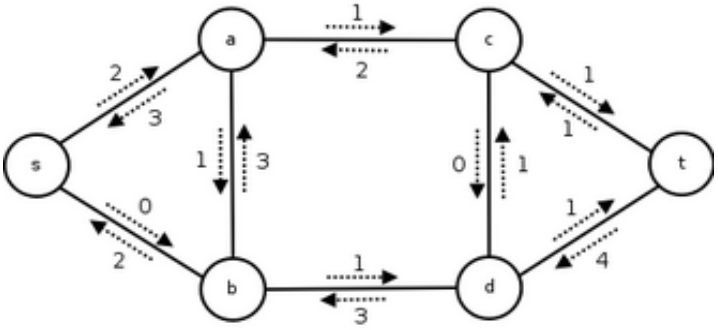
Ví dụ: Hình 1.9

Ở bên phải ta thấy một luồng trên mạng với nguồn có nhãn *s*, khối t và 4 nút được bổ sung. Lưu lượng và công suất được ký hiệu là *f/c*. Các mạng luôn đối xứng giới hạn dung lượng và dòng chảy bảo tồn. Tổng số lượng chảy từ *s* tới *t* là 5 có thể dễ nhìn thấy từ thực tế là tất cả luồng từ *s* là 5 đó cũng là dòng chảy đến *t*. Vậy trong bất kỳ 1 dòng chảy không tự nhiên xuất hiện hoặc không tự nhiên biến mất.



Hình 1.9. Một luồng trên mạng cho thấy lưu lượng và công suất dòng chảy [9].

Dưới đây ta thấy mạng còn dư cho dòng chảy nhất định, ta nhận thấy làm thế nào có công suất còn lại một số cạnh nơi công suất ban đầu là 0, ví dụ cho cạnh (*d, c*), dòng chảy này không phải là một luồng cực đại, dọc theo các đường dẫn (*s, a, c, t*), (*s, a, b, d, t*) và (*s, a, b, d, c, t*) là đường được thông nhau. Hướng kết nối còn lại là Min (*c(s, a)-f(s, a), c(a, c) - f (a, c), c (c, t) - f(c, t)*)= min (5-3,3-2, 2-1) = min(2,1,1) = 1. Nhận thấy hướng thông (*s, a, b, c, d, f*) không tồn tại trong mạng ban đầu, nhưng ta có thể gửi luồng dọc theo nó và vẫn nhận được một luồng hợp lệ.



Hình 1.10. Luông trên mạng hiển thị khả năng còn dư [9].

Nếu điều này là một mạng thực sự có thể là một dòng chảy của 2 từ *a* tới *b* và *a* mở ra cho 1 dòng chảy từ *b* tới *a*, nhưng ta chỉ duy trì dòng chảy ròng.

1.3. Một số khái niệm về Core

Core bên cạnh là các thành phần có cấu trúc kết nối từ các mạng lớn được phân chia thành những vùng mạng lưới nhỏ hơn dễ dàng trong việc xử lý các vùng của đồ thị lớn và mạng lưới lớn.

1.3.1. Khái niệm về Core, k-core

a. Cores [7]

Khái niệm về *Core* (*Core*) đã được giới thiệu bởi Seidman năm 1983.

Giả sử $G = (V, E)$ là một biểu đồ đơn giản, V là tập các đỉnh và E là tập hợp các dòng (các cạnh hoặc cung). Trong đó $n = |V|$ và $m = |E|$. Một đồ thị con $H = (C, E|C)$ biểu diễn bởi các bộ $C \subseteq V$ là một *K- core* (k-core) hoặc một *core* của bậc k nếu $\forall v \in C: deg_H(v) \geq k$ và H là một đồ thị con nhỏ nhất *Core* của một đồ thị nhỏ cũng được gọi là core chính. Số core của đỉnh v là thứ tự cao nhất của core. Khi đó điểm C là core tương ứng được xác định là *core* của H.

Trong đó *deg(v)* là bậc các trọng số của một đồ thị vô hướng được xác định bởi bên trong cung, ngoài cung hoặc cả trong cung và ngoài cung,… được xác định các loại khác nhau của core.

Các *Core* có các tính chất quan trong sau:

Các core được lồng nhau: $i < j \Rightarrow Hj \subseteq Hi$

Cores không nhất thiết phải kết nối với đồ thị con.

b. P-core [8]

Cho $N = (V, E, w)$ là một mạng lưới, khi đó $G = (V, E)$ là một đồ thị và $w: E \rightarrow IR$ là một điểm được gán cho dòng. Một hàm biến đổi trên đỉnh N hoặc một hàm P được gán với hàm $p(v, U), v \in V, U \subseteq V$ là một giá trị thực.

Ví dụ: về chức năng hữu đỉnh cho $N(v)$ biểu diễn tập hợp các lân cận của đỉnh v trong đồ thị G và $N(v,U) = \mathrm{N}(v) \cap U, U \subseteq V$.

(1) $p_1(v, U) = degU(v)$

(2) $p_2(v, U) = indegU(v)$

(3) $p_3(v, U) = outdegU(v)$

(4) $p_4(v, U) = indegU(v) + outdegU(v)$

(5) $p_5(v, U) = \sum_{u \in N(v,U)} \omega(v,u)$ khi $w : E \rightarrow \Re_0^+$

(6) $p_6(v, U) = \max_{u \in N}(v,U)\ w(v, u)$, khi $w : E \rightarrow \Re$

(7) p7(v, U) = số chu kỳ của của chiều dài k qua đỉnh v

Đồ thị con $H = (C, E|C)$ thiết lập bởi $C \subseteq V$ là một *p-core* của bậc $t \in \omega$. Nếu:

- $\forall v \in C : t \leq p(v, C)$
- C nằm trong bộ nhớ lớn nhất.

Chức năng của p không thay đổi nếu nó là thuộc 1 tính của V

$C_1 \subset C_2 \Rightarrow \forall v \in V : (p(v, C1) \leq p(v, C2))$

Tất cả các chức năng $p_1 – p_7$ đều đơn điệu.

Đối với chức năng đơn điệu *p-core* ở mức t bằng cách xác định xóa đỉnh có giá trị p thấp hơn t:

$C := V;$

Trong khi $\exists v \in C$: p(v, C) < t thì $C := C \setminus \{v\}$;

1.3.2. Tính chất của Core [7]

Tính chất 1: Cho chức năng đơn điệu p không thay đổi thủ tục ở trên *p-core* tại mức t.

Khi đó tập C đặt trở lại bởi các thủ tục có tính chất đầu tiên từ định nghĩa *p-core*.

Ta thấy rằng đối với dẫy đơn điệu p kết quả của thủ tục này là độc lập sau khi xóa thứ tự từng bước.

Trái với định lý, giả sử có hai *p-core* khác nhau ở mức độ *t* được xác định bởi bộ *C* và *D*. Core của *C* là tập đơn điệu nếu xóa trình tự u_2, u_3,... và *D* bằng chuỗi v_1, v_2, v_3,..., v_q. Ta có $D \setminus C = 0$ vậy dẫn đến mâu thuẫn.

Khi đó bất kỳ $z \in D\backslash C$. Để hiển thị cũng có thể bị xóa bỏ, áp dụng trình tự v_1, v_2, v_3,..., v_q để có được khi $z \in D\backslash C$ nó xuất hiện trong chuỗi u_1, u_2, u_3,..., = *z*. Hãy $U_0 = 0$ và $U_i = U_{i-1} \cup u_i\}$ khi $\forall i \in 1...p : p(u_i, (V \setminus D) \setminus U_{i-1}) < t$ vì vậy cũng là tất cả giao diện $u_i \in D\backslash C$ xóa $D\backslash C = \phi$ là mâu thuẫn.

Khi kết quả của các thủ tục được xác định duy nhất và đỉnh ở bên ngoài *C* có giá trị *p* thấp hơn *t*, *C* thiết lập cuối cùng thỏa mãn cũng điều kiện thứ hai từ định nghĩa của *p-core* là *p-core* mức thứ *t*.

Tính chất 2: Cho hàm đơn điệu *p* có các core lồng nhau

$t_1 < t_2 \Rightarrow H_{t2} \subseteq H_{t1}$

Chứng minh: Từ tính chất 1 kết quả là độc lập với thứ tự, đầu tiên ta xóa bỏ từng bước và xác đinh H_{t1}. Tiếp theo xóa bỏ một số đỉnh thêm từ H_{t2} vì vậy $H_{t2} \subseteq H_{t1}$

Ví dụ hàm đơn điệu *p*: Hãy xem xét các hàm *p* sau.

$$p(v,U) = \begin{cases} 0 & N(v,U) = \phi \\ \dfrac{1}{|N(v,U)|} \displaystyle\sum_{u \in N(v,U)} \omega(v,u) & \text{Ngược lại} \end{cases}$$

Khi đó: $\omega : E \to \Re_0^+$ trong mạng $N = (V,E,\omega)$, V = {*a, b, c, d, e, f*} .

E	(*a*:*b*)	(b:*c*)	(*c*:*d*)	(*b*:*e*)	(*e*:*f*)
ω	4	1	3	1	3

Ta nhận được các kết quả khác nhau tùy thuộc vào việc xóa các đỉnh b hoặc c (hoặc e).

Mạng ban đầu là một *p-core* ở mức 2. Áp dụng các thuật toán cho mạng hiện có ta có ba sự lựa chọn cho đỉnh đầu tiên sẽ bị xóa: *b, c* hoặc *e*. xóa *b,* sau khi loại bỏ đỉnh cô lập *a, p-core* C_1 = {*c, d, e, f*} ở mức độ 3. Lưu ý: các giá trị của *p* trong đỉnh c và e tăng từ 2 lên 3.

Xóa *c* (hoặc *e* đối xứng trong trường hợp đầu tiên được phân tích) ta nhận được C_2 = {*a, b, e, f*} ở mức độ 2 - giá trị tại *b* tăng lên đến 2,5. Trong bước tiếp

theo ta có thể xóa đỉnh *b*, thiết lập $C_3 = \{e, f\}$ ở mức 3 hoặc đỉnh *e* thiết lập *p-core* $C_4 = \{a, b\}$ ở mức 4.

Như ta đã thấy kết quả của các thuật toán phụ thuộc trên thứ tự xóa *p-core* ở mức 4 không được chứa trong *p-core* ở mức 3.

Như vậy với nội dung chương 1: Khai quát được tổng quan về lý thuyết đồ thị, một số định nghĩa cơ bản về đồ thị, các khái niệm liên quan đến mạng xã hội, phân tích mạng xã hội và một số khai niệm, tính chất liên quan về core để làm tiền đề cho việc tìm hiểu nội dung trong chương 2.

CHƯƠNG 2.
MỘT SỐ THUẬT TOÁN NHANH TÌM K-CORE TRONG MẠNG XÃ HỘI

K-core của một đồ thị là đồ thị con lớn nhất trong đó mỗi đỉnh được kết nối với ít nhất k đỉnh khác trong đồ thị con. Core phân hủy tìm *k - core* của đồ thị cho mỗi thực thể *k*. Nghiên cứu trước đây đã chỉ ra các ứng dụng quan trọng của core phân hủy như trong việc nghiên cứu các tính chất của các mạng lớn (ví dụ, tính bền vững, kết nối, trung tâm, v.v.), để giải quyết vấn đề *NP-khó* có hiệu quả trong các mạng thực tế (ví dụ, nhóm tối đa tìm kiếm, tính đồ thị con dày đặc nhất, v.v.), và trong liên kết mạng quy mô lớn, trực quan. *K-core* là một khái niệm được chấp nhận một phần vì có tồn tại một thuật toán đơn giản và hiệu quả để core phân hủy, đệ quy loại bỏ các đỉnh, các cạnh ở mức độ thấp nhất. Tuy nhiên, các thuật toán này yêu cầu truy cập ngẫu nhiên vào các đồ thị, do đó giả định toàn bộ đồ thị có thể được lưu giữ trong bộ nhớ chính. Mạng lưới thực tế như mạng xã hội trực tuyến đã trở nên cực kỳ lớn trong những năm gần đây và vẫn tiếp tục phát triển với một tốc độ ổn định. Ta đề xuất các thuật toán bên ngoài bộ nhớ đầu tiên để core phân hủy trong đồ thị lớn. Khi bộ nhớ đủ lớn để chứa đồ thị, thuật toán đạt được hiệu năng tương đương như các thuật toán trong bộ nhớ. Khi biểu đồ là quá lớn để được lưu giữ trong bộ nhớ, thuật toán chỉ yêu cầu có độ khó là $O(k_{max})$ quét của đồ thị, nơi k_{max} là số core lớn nhất của đồ thị. Ta chứng minh hiệu quả của thuật toán trên các mạng lưới lớn thực sự với lên đến 52,9 triệu đỉnh và 1,65 tỷ cạnh [5].

2.1. Thuật toán tìm *Cores* [7]

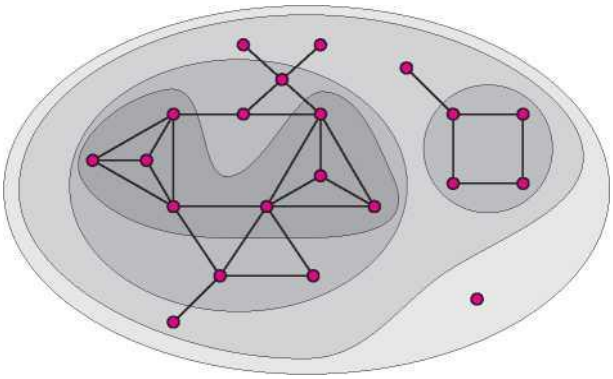
Cấu trúc của mạng lưới lớn có thể được tách ra bởi những phần vùng và chia chúng thành các bộ phận nhỏ, dễ sử dụng hơn. Đây là một trong những chủ đề được rất nhiều nhà khoa học tìm hiểu, nghiêm cứu. Thực vậy dựa trên k-core, đề xuất trong năm 1983 bởi Seidman. Trên báo cáo về một hiệu suất có độ phức tạp cao $O(m)$, m là số dòng, thuật toán cho việc xác định core phân hủy được đưa ra của một mạng.

Một trong những mối quan tâm lớn của phân tích mạng xã hội là xác định các nhóm con có kết cấu của các nút điểm trong cùng một mạng lưới. Nhóm cố

kết là tập hợp con của các nút điểm trong đó có những mối quan hệ tương đối mạnh mẽ, trực tiếp, cường độ cao, thường xuyên hoặc tích cực. Một số khái niệm đã được giới thiệu chính mô tả nhóm cố kết: các nhóm cliques, n-cliques, n-clans, n-clubs, k-plexes, k-cores, lambda... Đối với hầu hết các nhóm thuật toán có độ khó cao (NP khó có ít nhất ít bậc hai), nhưng đối với cores thì tồn tại một thuật toán rất hiệu quả.

2.1.1. Mô tả thuật toán

Cho $G = (V, E)$ là một đồ thị. V là tập hợp các đỉnh và E là tập các dòng (cạnh hoặc góc). Biểu thị $n = |V|$ và $m = |E|$, một đồ thị con $H = (W, E\backslash W)$ được ánh xạ bởi các điểm W là một *k-core* hoặc một *core* thuộc các bậc kiểu k khi đó $\forall(v) \in W$: $\deg_H(v) \geq k$ và H có giá trị thuộc tính lớn nhất của đồ thị. Bậc lớn nhất của core cũng là core chính. Số core của đỉnh v là thứ tự cao nhất của một core chứa đỉnh này.



Hình 2.1: 0, 1, 2 và 3 core phân hủy của một đồ thị [7].

Các bậc $deg(v)$ có thể là: bậc vào, bậc ra, bực vào + bậc ra… xác định loại core khác nhau.

Core lồng nhau: $i < j \Rightarrow H_j \subseteq H_i$

Core không nhất thiết phải liên kết trong đồ thị.

Mô tả thuật toán

Để xác định hệ thống core dựa trên các điểm sau:

Cho một đồ thị $G = (v, e)$ đệ quy xóa tất cả các đỉnh và các dòng, cạnh của các bậc bé hơn k đồ thị còn lại là k-core.

Phác thảo thuật toán như sau:

Thuật toán 2.1:

Đầu vào: Cho đồ thị $G = (v, e)$ hiển thị danh sách các lân cận xung quanh của điểm mút trong đồ thị.

Đầu ra: Tập core bằng số core của mỗi đỉnh.

1.1 Tính các gốc độ của đỉnh;

1.2 Tập hợp thứ tự các đỉnh V được tăng dần;

2 for mỗi $v \in V$ theo thứ tự do begin

2.1 *core[v] := diem[v];*

2.2 for *each u ∈ Neighbors(v)* do

2.2.1 *if degree[u] > degree[v] then begin*

2.2.1.1 *degree[u] := degree[u] - 1;*

2.2.1.2 *reorder V accordingly*

end;

end;

Trong thuật toán qua các bước của phép toán sàng lọc ta triển khai và hiệu quả thực hiện của các bước là 1.2 và 2.2.1.2.

Thuật toán:

Mô tả thuật toán bằng thuật ngữ Pascal.

Biểu đồ cơ cấu được sử dụng để đại diện cho đồ thị *G = (V, E).*

Về cấu trúc chi tiết trong thuật toán sẽ không được mô tả bởi vì có rất nhiều cách để giải quyết bài toán core trong đồ thị.

Giả sử các đỉnh của G được đánh số từ 1 đến n.

Các chức năng về kích cỡ và trong lân cận được mô tả ở bảng sau:

Tên (thông số)	**Giá trị trả lại**
Kích thước (*G*)	Số đỉnh trong đồ thị *G*
u nằm trong lân cận (*G*,*v*)	*u* không thuộc lân cận các đỉnh *v* trong đồ thị *G*

Bằng việc sử dụng một điểm lân cận thuộc đồ thị *G* (danh sách lân cận), ta có thể thực hiện cả hai chức năng để chạy trong thời gian liên tục.

Hai kiểu nguyên trong mảng (*tableVert* và *tableDeg*) cũng là trong lân cận của đồ thị. Một trong 2 *tableVert* và *tabeDeg* phải có độ dài ít nhất là *n*. Sự khác biệt duy nhất là làm thế nào chỉ ra các yếu tố trong danh sách các điểm lân cận, với chỉ số 1 ở *tableVert* và chỉ số 0 trong *tableDeg*.

Thuật toán được thực hiện bằng thủ tục *core*. Đầu vào là đồ thị *G* đại diện bởi loại biểu đồ có biến là *g*, đầu ra là mảng *deg* loại *tableVert* có chứa số core cho mỗi đỉnh của đồ thị *G*.

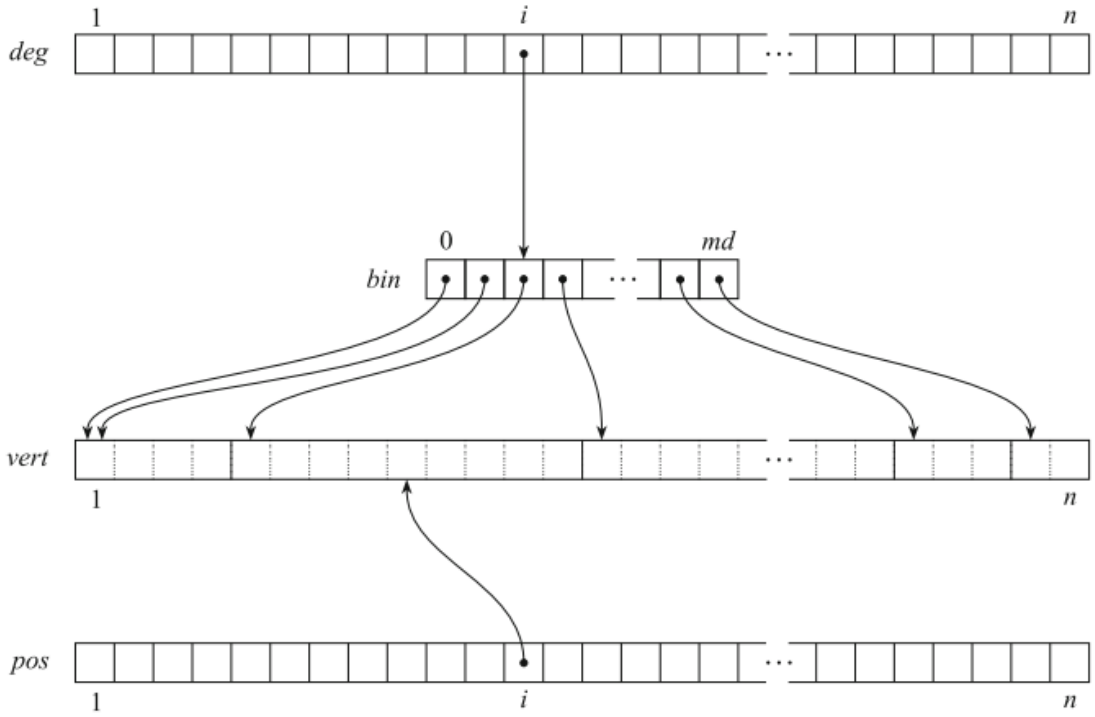
Giả sử ta dùng (03-06) một số biến số nguyên và bổ sung thêm 3 vào mảng. Mảng *vert* chứa tập các đỉnh được sắp xếp từng mức độ. Vị trí của đỉnh trong mảng đỉnh sẽ được lưu ở các điểm trong mảng bộ chứa trong mảng.

Thuật toán 2.2: Thuật toán đơn giản core trong các đồ thị.

```
0   procedure cores(var g: graph; var deg: tableVert);
1           var
2               n, d, md, i, start, num: integer;
3               v, u, w, du, pu, pw: integer;
4               vert, pos: tableVert;
5               bin: tableDeg;
6   Begin
7       n := size(g); md := 0;
8       for v := 1 to n do begin
9           d := 0; for u in Neighbors(g, v) do inc(d);
10          deg[v] := d; if d > md then md := d;
12      end;
13     for d := 0 to md do bin[d] := 0;
14      for v := 1 to n do inc(bin[deg[v]]);
15          start := 1;
16     for d := 0 to md do begin
17          num : = bin [d] ;
18          bin[d] := start;
19          inc(start, num);
```

```
20          end;
21          for v := 1 to n do begin
22              pos[v] := bin[deg[v]];
23               Vert [pos [v]] :=v;
24               inc(bin[deg[v]]);
25           end;
26           for d := md downto 1 do bin[d] := bin[d-1] ;
27                bin[0] := 1;
28          For i := 1 to n do begin
29               v := vert [i] ;
30              for u in Neighbors (g, v) do begin
31                   if deg[u] > deg[v] then begin
32                      du := deg[u] ; pu := pos [u] ;
33                       pw : = bin [du] ; w := vert [pw];
34                       if u <> w then begin
35                                  pos[u] := pw; vert[pu] := w;
36                                  pos[w]:= pu; vert[pw] := u;
37                        end;
38                            inc(bin[du]); dec(deg [u]);
39                   end;
40               end;
41           end;
42  end;
```

Chức năng	Ý nghĩa
Dec(a)	A := a – 1
Inc(a)	A : a + 1
Inc(a, b)	A := a + b



Hình 2.2: Mảng truyền dữ liệu [7].

Mỗi một điểm có thể nằm ở vị trí của đỉnh kế tiếp điểm đầu tiên của các đỉnh trong mảng. Xem hình 1.9 việc dùng thuật ngữ Pascal thực hiện một thuật toán cho ta thấy các trường hợp đơn giản của đồ thị *G = (V, E)* vô hướng, trong đó *E* là tập hợp của các cạnh.

Trong một kiểm nghiệm thực tế các thuật toán được sử dụng tự động dùng cho mảng. Để đơn giản hóa ta mô tả thuật toán bằng những biện pháp thay thế các chuỗi có chủ đích.

Bước đầu phải khởi tạo một số biến lân cận và các mảng (08-12). Xác định *n là* số lượng các đỉnh của đồ thị *g*. Sau đó tính mỗi đỉnh *v* trong đồ thị *g* và lưu nó vào mảng *core* đồng thời cũng tính bậc tối đa cho *md*. Sắp xếp các đỉnh tăng dần bằng cách phân loại các *bin* thứ hạng (dòng 13-25).

Bước 1: Đếm (dòng 13-14) có bao nhiêu đỉnh sẽ nằm trong mỗi *bin* (*bin* bao gồm đỉnh với cùng một mức). Bộ chứa được đánh số từ *0* đến *md*.

Từ kích thước bộ chứa được xác định (dòng 15-20) bắt đầu từ vị trí của bộ chứa trong mảng *vert*. Bin *0* bắt đầu tại vị trí *1* trong khi bộ chứa bắt đầu tại vị trí tương đương với tổng bằng kích thước của các bộ chứa trước đó. Để tránh việc

các mảng trùng nhau ta chỉ sử dụng cùng một mảng (*bin*) để lưu trữ các vị trí bắt đầu của bộ chứa.

Bước 2: đưa đỉnh (dòng 21-25) của đồ thị *G* vào mảng *vert*. Mỗi đỉnh *bin* nó thuộc về vị trí bắt đầu của bộ chứa. Vì vậy ta có thể đưa đỉnh đến nơi thích hợp, nhớ vị trí của nó trong bảng *pos* và tăng vị trí bắt đầu của bộ chứa được sử dụng. Các đỉnh bây giờ được sắp xếp với mức độ tăng dần.

Bước đầu khởi tạo giai đoạn cuối cùng ta phải khôi phục lại vị trí bắt đầu của các bộ chứa (dòng 26-27). Tăng các bước lên nhiều lần khi đặt đỉnh vào bộ chứa tương ứng. Thay đổi vị trí ban đầu của bộ chứa tiếp theo, để khôi phục lại phải bắt đầu từ từng vị trí di chuyển các giá trị trong mảng *bin* cho một vị trí bên phải. Đặt lại vị trí bắt đầu của *bin* 0-1 có giá trị phân hủy core, triển khai thực hiện cho mỗi vòng lặp từ các thuật toán được mô tả trong phần 3, được thực hiện trong vòng lặp chính (dòng 28-41) chạy qua tất cả đỉnh *v* trong đồ thị *g* theo thứ tự, được xác định bởi bảng *vert*. Số *core* hiện tại đỉnh *v* là mức độ hiện tại của đỉnh đó. Con số này đã được lưu trữ trong bảng. Cho mỗi lân cận *u* của đỉnh *v* với mức độ cao hơn và di chuyển nó cho một *bin* ở bên trái. Di chuyển các đỉnh *u* cho một *bin* bên trái.

2.1.2. Đánh giá độ phức tạp của thuật toán

Các thuật toán mô tả chạy trong thời gian $O(max(m,n))$. Để tính toán tất cả các đỉnh (dòng 08-12) ta cần thời gian $O(max(m, n))$ xét mỗi dòng ít nhất 2 lần chạy. Câu lệnh *Bin sort* thực hiện bước (dòng 13-27) chạy cho 5 vòng đến kích thước *n* lần với thời gian liên tục O(1) do đó nó chạy trong thời gian O(*n*).

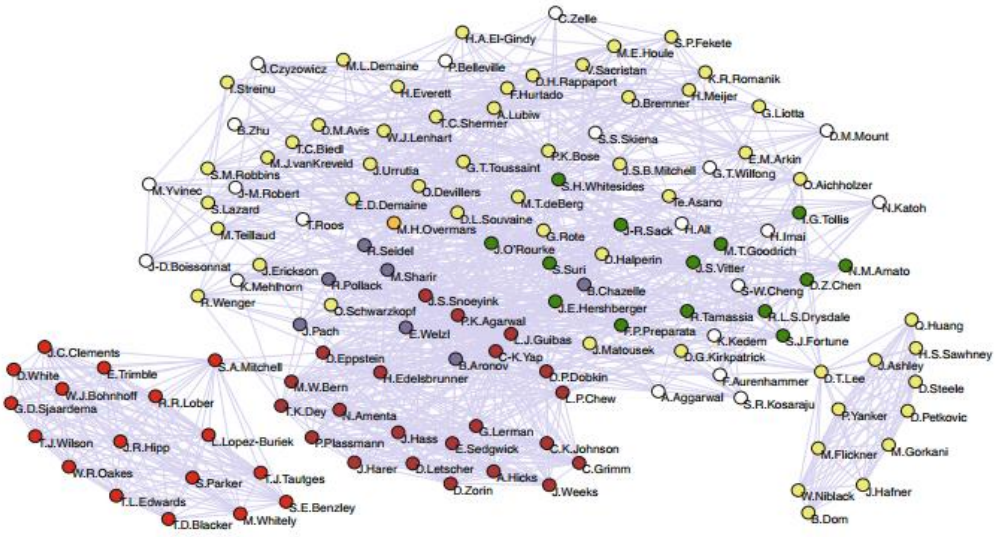
Kết quả thuật toán đánh giá (dòng 29) cần một thời gian liên tục có độ khó O(*n*). Các câu lệnh có điều kiện (dòng 31-39) cũng chạy trong thời gian thực. Mỗi cạnh của *G* được thực hiện 2 lần hầu hết trong vòng lặp (dòng 30-40), trong tất cả các lần lặp (dòng 28-41) có độ khó là $O(max(m, n))$

Đánh giá độ phức tạp của thuật toán: Thuật toán có độ khó là $O(max(m, n))$, trong một kết cấu mạng $m \geq n-1$ và do đó $O(max(m, n)) = O(m)$.

2.2. Thuật toán tìm p-core [8]

2.2.1. Hàm đơn điệu p và core

Khái niệm "core" dựa trên khái niệm bậc cổ điển được tổng quát để xem xét các đặc tính khác của đỉnh (nhóm). Trong phân tích một mạng lưới, việc tiếp cận để xác định các yếu tố quan trọng hoặc các bộ phận của mạng lưới được thể hiện bởi tầm quan trọng của một đỉnh là một hàm thuộc tính đỉnh *p*.



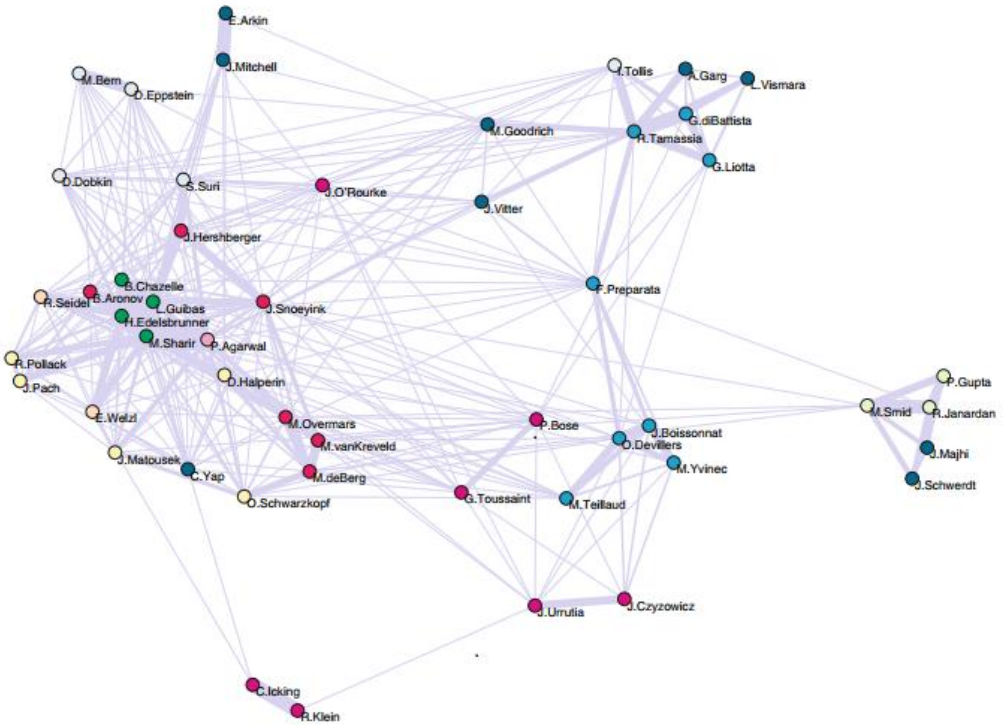
Hình 2.3: Core trong mạng được phân tích bằng hình học [8].

2.2.2. Một số ví dụ về hàm đơn điệu p

Ví dụ: Ta hãy xem lại các mạng trong không gian hình học. Việc thay thế nhiều cặp cạnh ở giữa bằng một cạnh đa dạng và duy nhất, thông thường các đỉnh thuộc các hàm hữu hạn $p_1 = deg$, các *core* tương ứng p_1-*core* cho ta câu trả lời: Đâu là nguồn cung mức với dữ liệu lớn, và các vị trí kề được xác định mục đích như thế nào. Để xác định câu trả lời: ta thích hợp sử dụng chức năng cho đỉnh. Chức năng $p_S \equiv p_{11}$ trong đó tổng số công việc là nguồn cung mức từ core và các core từ nghiên cứu khác. Thu thập thông tin trên mạng, áp dụng thuật toán 4 để xác định p_S-*core*. Hình 2.4 cho thấy p_S-*core* ở 46 mức. Ở mức này được xác định bằng cách xem xét sự phân bố tần số của các giá trị trong hàm p_S. Các trọng số dòng được đại diện bởi độ dày của các điểm lân cận.

Các p_S-*core* cho ta ứng dụng tích hợp ở lĩnh vực tính toán trong hình học. Một số *deg-core* chính cũng thuộc vào p_S-*core*. Bên cạnh những nhóm chính ở phía bên trái một số cặp xuất hiện: M. Bern, D. Eppstein; E. Arkin, J. Mitchell; J.

Urrutia, J. Czyzowicz; P. Bose, G. Toussaint; C. Icking, R. Klein; và ba nhóm: Tamassia của nhóm (R. Tamassia, G. Liotta, I. Tollis, A. Garg, G. diBattista, L. VISMARA, M. Goodrich, J. Vitter), nhóm của Canada (M. Teillaud, M. Yvinec , O. Devillers, J. Boissonnat) và nhóm Smid của (M. Smid, P. Gupta, J. Majhi, R. Janardan, J.Schwerdt). F. Preparata là một loại trung gian giữa của Tamassia và nhóm của Canada.



Hình 2.4: Ps-core trong mạng mô phỏng bằng hình học được tính toán ở 46 mức[8].

2.2.3. Core tổng quát và tính chất.

Cho $N = (V, E, w)$ là một mạng trong đồ thị $G = (V, E)$ và w: $E \rightarrow$ R là một chức năng được gán giá trị cho dòng. A nằm trên đỉnh N, hoặc một hàm *p-function* ngắn, có chức năng hàm $p(v, C), v \in V, C \subseteq V$ bằng giá trị thực. Xác định tập con C là một phần của mạng lưới có giá trị $p(v, C)$ được tính.

Cho $N(v)$ tập hợp biểu thị các điểm lân cận của đỉnh v trong đồ thị G,

$N+(v) = N(v) \cup \{v\}$, và $N(v, C) = N(v) \cap C$ của một đỉnh thuộc $C \subseteq V$.

Ví dụ sau chứng minh hàm đơn điệu các hàm p_1, p_2, p_3 và p_4 là hàm đơn điệu trong core.

– $p_1(v, C) = \deg(v, C)$

– $p_2(v, C)$ = indeg(v, C)

– $p_3(v, C)$ = outdeg(v, C)

– $p_4(v, C)$ = indeg(v, C) + outdeg(v, C)

– $p_5(v, C) = \frac{\deg(v,C)}{\deg(v)}$, nếu deg(v) > 0; = 0, mặt khác

– Mật độ tương đối

- $p_6(v, C) = \frac{\deg(v,C)}{\max_{u\in N(v)} \deg(u)}$, nếu deg(v) > 0; = 0, mặt khác

– Sự đa dạng của lân cận.

$p_7(v, C) = \max_{u\in N(v,C)} \deg(u) - \min_{u\in N(v,C)} \deg(u)$

– Sự đa dạng trong các vùng lân cận

$p_8(v, C) = \max_{u\in N+(v,C)} \deg(u) - \min_{u\in N+(v,C)} \deg(u)$

– Hệ số nhóm: Cho đồ thị vô hướng $G = (V, E)$, $K(U)$ biểu thị các biểu đồ trên một tập đỉnh $U \subseteq U$ và $E(N)$ là tập các cạnh E nằm trong khoảng giữa các đỉnh của N.

$p_9(v,C) = \frac{|\varepsilon(n(v,C))|}{|\varepsilon(K(N(v)))|}$, nếu deg(v) > 1; = 0 mặt khác

– Các cụm điều chỉnh hệ số phân nhóm: $\Delta(G) = \max_{u\in v} \deg(u)$

$p_{10}(v, C) = \frac{\deg(v)}{\Delta(G)} p_9(v,C)$

– $p_{11}(v,C) = \sum_{u\in N(v,c)} \omega(v,u)$, khi $\omega: E \to R_0^+$

– $p_{12}(v,C) = \max_{u\in N(v,C)} \omega(v,u)$, $Khi\ \omega: E \to R$

– $p_{13}(v,C)$ = số chiều dài vòng lặp k của đỉnh v quét qua các đỉnh từ C

– Giá trị trung bình $p_{14}(v,C) = \frac{1}{|N(v,C)|} \sum_{u\in N(v,C)} \omega(v,u)$, nếu $N(v, C) \neq \emptyset$; = 0

2.2.4. Thuật toán tìm p-core

Cách đơn giản nhất để xác định các phần tử của một mạng là sử dụng đỉnh được cắt ở ngưỡng t đã chọn.

- Các bộ phận tương ứng như việc kết nối các thành phần liên kết bởi mạng con trên việc thiết lập của các đỉnh $C \subseteq V$ bằng giá trị $p(v) \geq t$.

$C = \{v \in V: p(v) \geq t\}$

Lưu ý: Các giá trị này không cần phải có ít nhất t nếu p bị giới hạn để thu được mạng con. Yêu cầu này dẫn đến khái niệm của *p-core*.

Các đồ thị con $H = (C, E|C)$ được thiết lập bởi các đỉnh $C \subseteq V$ gọi là một *p-core* ở mức độ $t \in \Re$

– $\forall v \in C: t \leq p(v, C)$

– C có giá trị thuộc tính lớn nhất.

Lưu ý: Thông thường một *k-core* là một p_1-*core* ở mức k

p là một dẫy đơn điệu khi và chi khi nó cũng là một thuộc tính

$C_1 \subset C_2 \Rightarrow \forall v \in V: (p(v, C_1) \leq p(v, C2))$

Trong các từ, giá trị $p(v, C)$ của của hàm hữu hạn đỉnh p cho một đỉnh v sẽ không giảm nếu ta tăng tập con $C \subseteq V$.

Tất cả các hàm p_1,...., p_{13} là đơn điệu, hàm p_{14} cũng đơn điệu. Cho một vài hàm đơn điệu *p-function* được sử dụng thông thường để mở rộng các core, tới trường hợp của *p*-cores. Khi *p*-core Ht là bậc t được xác đinh bởi số lần xóa các đỉnh của p thấp hơn giá trị so với t.

$C := V$; **while** $\exists v \in C : p(v, C) < t$ **do** $C := C \setminus \{v\}$;

Định lý 2.1. Đối với mỗi hàm đơn điệu, chức năng hữu hạn ở đỉnh p của thuật toán 2 được xác định bởi *p-core* trong đó H_t ở mức t.

Chứng minh: Tập C có tính chất đầu tiên được xác định từ việc thủ tục *p-core* xóa các điểm kề. Ta cũng thấy rằng đối với hàm đơn điệu p cho kết quả độc lập khi thứ tự các điểm được xóa.

Giả sử, ngược lại có hai mức khác nhau *p-core* ở mức t, xác định bởi C và D. Khi đó core C là một thủ tục xóa trình tự từ các điểm $u_1, u_2, u_3, \ldots, u\ p$; và D bởi các điểm $v_1, v_2, v_3, \ldots, vq$. Giả sử $D \setminus C \neq \emptyset$. Dẫn đến mấu thuẫn.

Vì $z \in D \setminus C$. cũng có thể xóa các điểm đầu tiên. Ta áp dụng xóa trình tự $v_1, v_2, v_3, \ldots, vq$ nhận được D. Từ $z \in D \setminus C$ xuất hiện trong chuỗi $u_1, u_2, u_3, \ldots. s = z$. Từ $U_0 = \emptyset$ và $Ui = Ui{-}1 \cup \{ui\}$.

Khi đó xuất hiện trong tất cả $p(ui, V \setminus Ui-1) < t$

$i \in 1 \ldots p$, ta có các hàm đơn điệu của p, cũng xuất hiện trong tất cả các $p(ui,(V \setminus D) \setminus Ui-1) < t$.

$i \in 1 \ldots p$. Do đó cũng xuất hiện trong tất cả $ui \in D \setminus C$ khi chúng được xóa $D \setminus C = \emptyset$ - a => mâu thuẫn.

Kết quả quá trình được xác định bên ngoài các đỉnh C có p có giá trị thấp hơn t nhiều, ở mức xác định từ *p-core* cũng là *p-core* ở mức độ *t.*

Kết quả 1: Đối với mỗi hàm p, chức năng p không thay đổi khi được lòng nhau.

$t_1 < t_2 \Rightarrow Ht_2 \subseteq Ht_1$

Giả sử: Theo định lý 1. Ta có kết quả thứ tự được xác định Ht_1, tiếp theo xóa một vài thủ tục chức năng trên đỉnh Ht_2. Bởi vì $Ht_2 \subseteq Ht_1$.

Cho một hàm đơn điệu các dẫy số p xét những mạng lưới $N = (V, E,w)$ bằng tập đỉnh $V = \{a, b, c, d, e, f\}$ và trọng lượng w: $E \rightarrow \Re_0^+$

L	$(a : b)$	$(b : c)$	$(c : d)$	$(b : e)$	$(e : f)$
w	4	1	3	1	3

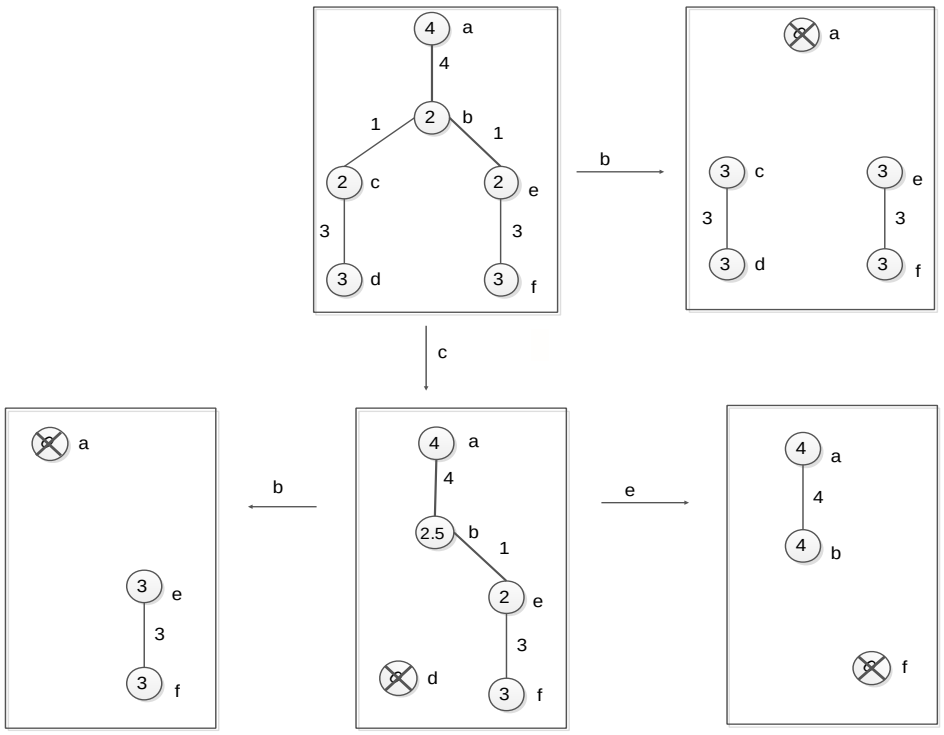
Trong hình 2.3. Đối với p-cores xác định ta sử dụng hàm chức năng *p-function,* p_{14} không sử dụng hàm đơn điệu.

Cho ví dụ, $\{c, d\} \subset V$, nhưng $p_{14}(c, \{c, d\}) = 3$ và $p_{14}(c, V) = 2$.

Trong hình 2.3 Giá trị của *p-function* được hiển thị bên trong đỉnh của vòng tròn, các cạnh được định tuyến với cùng trọng số tương ứng những mũi tên giữa các phần khác nhau của mạng lưới đều được đánh dấu khi đỉnh bị xóa Các mạng ban đầu là một p_{14}-*core* ở mức độ 2. Áp dụng thuật toán 2 vào mạng ta có các đỉnh đầu tiên cần được xóa: *b, c* hoặc *e*. sẽ bỏ đỉnh *b* mà ta nhận được. Sau khi loại bỏ các đỉnh cô lập *a* của *p-core* $C_1 = \{c, d, e, f\}$ ở mức 3.

Lưu ý: Các giá trị của p trong đỉnh c và e tăng từ 2 đến 3.

Bỏ đỉnh c (hoặc đỉnh đối xứng e – ta có trường hợp đầu tiên) ta có được những bộ $C_2 = \{a, b, e, f\}$ ở mức 2 – và giá trị tại b tăng lên 2.5. Trong bước tiếp theo xóa các đỉnh *b,* thiết lập $C_3 = \{e, f\}$ ở mức 3, hoặc đỉnh *e,* thiết lập p-core $C4 = \{a, b\}$ ở mức 4.



Hình 2.5: Thứ tự được xóa biểu diễn trong hàm đơn điệu p-core [8].

Kết quả nhận được phụ thuộc vào thứ tự xóa trên các đỉnh *b* hoặc *c* (hay *e*) các *p-core* ở mức 4 không được chứa trong *p-core* ở mức 3.

Thuật toán của *p-core* ở mức bậc *t.* Một hàm *p-functions* được gọi là lân cận khi và chỉ khi $p(v, C) = p(v, N(v, C))$, của tất cả $v \in V$ hoặc, giá trị của *v* chỉ phụ thuộc vào đỉnh *v*. Tất cả các hàm *p-functions* từ các lân cận ngoại trừ p_{13} cho $k \geq 4$. Từ ví dụ trên. Ta giả định cho chức năng *p* có cùng tồn tại một p_0 liên tục mà cho tất cả $v \in V$, $p(v, \emptyset) = p_0$.

Giả sử $p(v, N(v, C))$ có thể được tính trong $O(\deg(v, C))$ thời gian, ta thấy chức năng của một hàm đơn điệu và bộ thuộc tính đỉnh của hàm *p* được tồn tại trong thuật toán 3 xác định *p-core* ở mức *t* trong $O(m*\max(\Delta, \log n))$ thời gian.

Thuật toán 2.3:

Đầu vào: Cho đồ thị $G = (V, E)$ và những điểm thuộc danh sách lân cận, $t \in \Re$, chức năng của *p* không thay đổi

Đầu ra: $C \subseteq V$ Khi *C* là *p*-core của đồ thị *G* mức độ *t*.

1 $C := V$;

2. for $v \in V$ do $p[v] := p(v, N(v, C))$;

3. *build_min_heap(v, p)*;
4. while *p*[*top*] < *t* do begin
4.1. *C* := *C* \ {*top*};
4.2. for *v* ∈ *N(top, C)* do begin
4.2.1. *p*[*v*] := *p(v, N(v, C))*;
4.2.2. *update_heap(v, p)*;
end;
end;

Bước 4.2.1. Tăng tốc khi cập nhật *p*[*v*] và thay đổi từng bước trong vòng lặp của *p*[*v*].

Hệ thống bậc của *p-cores* Trong thuật toán 3 xác định *p-core* của đồ thị *G* ở mức độ *t* có thể phân mức và mở rộng (trong tập các đỉnh) của *p*-cores. Các core phân mức được xác định trong hệ thống gán cho mỗi đỉnh là giá trị cao nhất cho tất cả các *p-core* có chứa đỉnh.

Thuật toán 2.4:

Đầu vào: Đồ thị *G = (V, E)* cho danh sách các điểm lân cận tại đỉnh *p* thuộc tính chức năng không thay đổi.

Đầu ra: Bảng *core* bằng tập các core cho mỗi đỉnh.

1. *C* := *V*;
2. for *v* ∈ *V* do *p*[*v*] := *p(v, N(v, C))*;
3. *build_ min_heap(v, p)*;
4. while *sizeof(heap)* > 0 do begin
4.1. *C* := *C* \ {*top*};
4.2. *core*[*top*] := *p*[*top*];
4.3. for *v* ∈ *N(top, C)* do begin
4.3.1. *p*[*v*] := max { *p*[*top*], *p(v, N(v, C))*};
4.3.2. *update_heap(v, p)*;
end;
end;

Định lý 2.2. Giả sử $p(v, N(v, C))$ có thể được tính trong $O(\deg(v, C))$ thời gian sau đó cho các hàm đơn điệu và tập các đỉnh p ở thuật toán 4 xác định phân mức *p-core* trong $O(m*\max(\Delta, \log n))$ thời gian. Ta giả định P là thời gian tối đa cần thiết cho việc xử lý giá trị $p(v, C)$, cho tất cả $v \in V, C \subseteq V$. Khi đó độ phức tạp của các bước trong thuật 4 là $T_{1\text{-}3} = O(n) + O(P_n) + O(n \log n) = O(n \cdot max(P, \log n))$. Ta thấy tại vòng lặp *while*. Mỗi lần lặp lại tăng kích thước của bộ chứa khi đó số lần lặp C được giảm đi 1 cho những lần lặp n. Ở câu lệnh 4.1 và 4.2 có thể chạy với thời gian liên tục góp phần tăng các vòng lặp trong $T_{4.1,\,4.2} = O(n)$. Trong tất cả các vòng lặp và kết hợp với thời gian thực cho mỗi dòng được lặp lại ít nhất 1 lần. Do đó các vòng lặp trong (câu lệnh 4.3.1 và 4.3.2) thực hiện ít hơn m thơi gian – góp phần ít hơn $T_{4.3} = m(P + O(\log n))$ cho đến vòng lặp *while*. Các giá trị $p(v, N(v, C))$ được cập nhật trong thời gian liên tục - $P = O(1)$. Tổng hợp lại ta có độ phức tạp của thuật toán $T = T_{1-3} + T_{4.1,\,4.2} + T_{4.3} = O(m \cdot max(P, \log n))$. Cho mỗi hàm lân cận *p-function*, mà giá trị của $p(v, N(v, C))$ được tính là $O(\deg(v, C))$, ta có $P = O(\Delta)$.

2.3. Thuật toán tìm k-core địa phương [10]

Một mạng lưới được tạo thành bởi các đỉnh (hay các nút), và cạnh kết nối giữa các đỉnh, trong thế giới thực có hàng triệu các hệ thống khác nhau được mô tả như những hình thức mạng (mô tả trong lý thuyết đồ thị - toán học).

Ví dụ: bao gồm mạng xã hội, mạng sinh hoc, mạng Internet, mạng cộng đồng, các trang Web trên toàn thế giới…

Trong các nghiên cứu gần đây việc phân tích cơ cấu mạng như bước đột phá ở mọi khía cạnh của khoa học. Bình thường cách phân tích mạng để tìm việc đồng nhất giữa các đỉnh trong mạng và nhóm chúng với nhau sau đó chia thành những nhóm mang tính chất riêng biệt. Quá trình phân tích như trên được gọi là biểu đồ liên kết nhóm hoặc lân cận. Phân tích cấu trúc mạng điển hình chỉ tập trung vào việc phân tích giữa các cạnh và đỉnh trong đồ thị nhỏ. Trong những năm gần đây việc phát triển với quy mô lớn mang tính thống kê của mạng làm tiền đề mới cho việc phát triển trong các cuộc nghiên cứu mạng. Ngày nay công nghệ với những cỗ máy siêu máy tính đã giúp cho việc thu thập và phân tích dữ liệu lớn được dễ dàng hơn.

Ở thuật toán *k-core* lân cận ta đi tìm hiểu và phân tích những kết quả mà các nhà nghiên cứu thu thập được từ việc lấy dữ liệu và xây dựng để phân tích cấu trúc cộng đồng hay các lân cận ở mức độ *k-core*.

2.3.1. Mô tả thuật toán

Như việc mô tả, cấu trúc cộng đồng không phải là duy nhất mà các nhà nghiên cứu dựa trên xác định các thuộc tính và mật độ. Tập các đỉnh trong mạng được kết nối với nhau do đó chúng có sự tương đồng mạnh mẽ trong các lân cận khác. Việc phân tích mạng xã hội được xem như là những đồ thị nhỏ và tính chất của chúng là những chuỗi lân cận các đỉnh kề nhau.

Khái quát chung về k-core địa phương:

Các thuật toán *k-core* phân hủy tập trung vào mức độ của đỉnh, trong khi cấu trúc cộng đồng mô tả chi tiết về "mật độ" bên trong của đồ thị. Các khu vực của một đỉnh *v* là tập hợp của tất cả các đỉnh khác có kết nối trực tiếp với nó. Vùng lân cận là điều kiện của việc kết nối các đỉnh lân cận, nó còn là một thuộc tính cho việc nhận diện của cộng đồng. Trong thuật toán không xem xét các điều kiện là một đỉnh có khả năng thuộc các cộng đồng khác nhau. Thuật toán chỉ tập trung vào việc phân tích các đồ thị vô hướng và mạng. Dựa vào mức độ cộng đồng địa phương khái niệm k-core được đưa ra. Chứng minh mối quan hệ giữa các địa phương *k-core* và *k-cliqe* kết nối giữa các địa phương *k-core* và *k-core*. Thuật toán k-core địa phương được trình bày để mô tả tốt hơn các cấu trúc cộng đồng mạng.

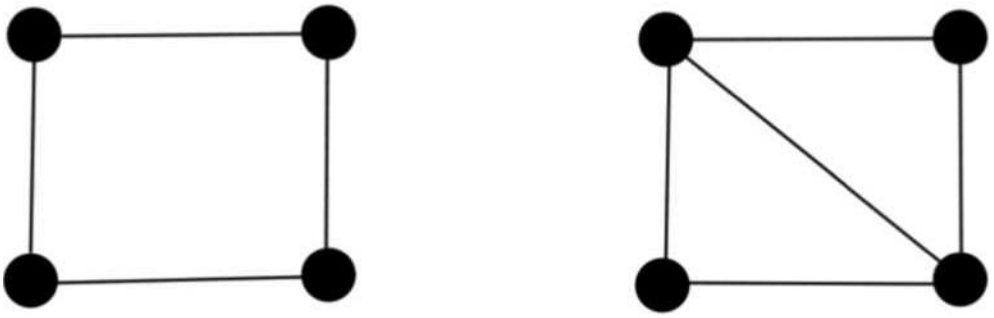
Trước tiên ta nhắc lại khái niệm về đồ thị: Một đồ thị *G* bao gồm tập các đỉnh *V*(*G*) và các cạnh *E*(*G*). Cho đồ thị vô hướng khi các cạnh được sắp xếp thuộc đỉnh *V*(*G*) nếu hai đỉnh đều kết thúc của một cạnh liền kề được gọi là lân cận. Mức đỉnh trong một đồ thì là số cạnh liền kề.

Một phân vùng *k-core* là một đồ thị con *G'* của *G* trong đó mỗi đỉnh của *G'* có các lân cận lớn hơn hoặc bằng *k* trong đồ thị con. Khi đó thuật toán *k-core* dựa trên các kết nối của đồ thị mà không xem xét cấu trúc các lân cận. Các lân cận của đồ thị con *k-core* trong thuật toán đưa ra một rạng của đồ thị con *k-core,* và chúng được so sánh như sau:

Định nghĩa 2.1. Một lân cận *k-core* là một đồ thị con *G'* của *G* mà mỗi đỉnh của *G* có cộng đồng địa phương ít nhất *k* trong đồ thị con khi và chỉ khi $min(V_{Ej}) \geq k. (j = 1,2, ..., n)$. Tương tự số *k-core* địa phương của một đồ thị con là *k*. Ta sử dụng K_E (*k*) cho *k-core* địa phương.

Định nghĩa 2.2. Một đỉnh *i* có địa phương *k-core* số địa chỉ *k* nếu nó thuộc về *k-core* địa phương nhưng không đến (*k + 1*) core; nó cũng có địa phương *k –core* số *k*. Các địa phương lớn nhất *k-core* có chỉ số *k* mà số *k-core* địa phương tối đa được ký hiệu là $K_E^{\max}$. Ta sử dụng $K_E(K_E^{\max})$ để biểu thị các đồ thị *con k-core* địa phương bằng số *k-core* địa phương tối đa.

Trên thực tê một *k – core* địa phương là phù hợp hơn cho cấu trúc cộng đồng trên *k-core,* mà một đồ thị được liên kết trong cộng đồng. Ta minh họa một ví dụ *k-core* địa phương trong Hình 2.6. Hình 2.6 là ví dụ cho ít nhất 4 – đỉnh 2 – *core* và 2 - core địa phương. Hình 2.6 (*a*) là có ít nhất 4 đỉnh 2 – core. Hình 2.6 (*b*) có ít nhất 4 đỉnh 2-*core* địa phương và ta có thể dễ dàng thấy rằng *k-core* địa phương có cạnh lớn hơn *k-core* và có các đỉnh kề nhau. Trong thực tế, 4-đỉnh 2 - core bằng với 4 đỉnh địa phương *e* – core. Từ định nghĩa của *k-core* địa phương, ta có các định lý sau:



(a) K-core có số core băng 2 ***(b) nhóm K-core có số core bằng 2***

Hình 2.6: k-core vs k-core lân cận; số lượng tối thiểu 2 core là 4 đỉnh và 4 cạnh; số lượng lân cận tối thiểu là 2 core 4 đỉnh bằng 5 cạnh [10].

Định lý 2.3. Khi *k = 2*, K_E (2) là một tập hợp các hình tam giác (3-*clique).*

Chứng minh: khi *n = 2*, kích thước của thành phần kết nối đỉnh trong đồ thị có ít nhất là 2; Vì vậy, các thành phần kết nối địa phương và đỉnh xây dựng một hình tam giác (*3 -clique*). Vì mỗi đỉnh trong thành phần kết nối địa phương

cũng có mức độ cộng đồng địa phương ít nhất 2, ta cũng có thể tạo thành hình tam giác K_E (2) là một tập hợp các hình tam giác.

Định lý 2.4. Khi *k*> *2*, K_E (*k*) là một tập hợp thấm *3-clique*.

Chứng minh: Từ định lý 1. Ta biết rằng K_E (2) là tập hợp các hình tam giác (*3 -clique*). Giả sử mỗi thành phần kết nối địa phương trong lân cận đỉnh của đồ thị có P_i; tạo nên P_i *- 1* hình tam giác, trong đó chia P_i - 2 cạnh. Từ định nghĩa về *clique*, những hình tam giác thấm *3-clique* nhóm với kích thước P_i + 1 vì vậy, hoàn toàn K_E(*k*) là một tập hợp thấm *3-clique*.

Định lý 2.5. Gọi *S*(*n*) là tập *n*-đỉnh ta có *S(n)* = K_E (*n* - 1).

Chứng minh: *S*(*n*) là một *clique* nhóm *n*-đỉnh, theo định nghĩa của *k-core* địa phương, tất cả các đồ thị thuộc lân cận đỉnh của *S*(*n*) được kết nối với các thành phần có kích thước là *n*-1; vậy *n*-đỉnh *clique* là tương đương với địa phương *n-1-core*.

Từ định lý ta chứng minh sự khác biệt giữa các *k-core* và địa phương *k-core*. Các địa phương *k – core* thực sự là dựa vào hình tam giác (hoặc *3 -clique*). Để cho phép *P*(*k*) (*k* > 3) là một *K – clique* thấm. Theo định nghĩa của nhóm *clique* thấm và định lý 3, ta dễ nhận biết *P*(*k*) = K_E (*k* - 1). Từ khi *P*(*k* + 1) $\subseteq$ *P*(*k*), K_E(*k* - 1) ở đây cũng là bộ thấm *3-clique.*

Tại sao ta sử dụng các hình tam giác (hoặc 3 *-clique* như là một tham số cho *k-core* địa phương ?. Trước tiên, các tam giác là một tham số quan trọng đối với hệ số phân nhóm, còn là một đặc tính có nguyên tắc trong đồ thị; thứ hai, từ các loại định nghĩa của cộng đồng, *clique* đóng một vai trò quan trọng trong cấu trúc. Ví dụ nhiều nhất cho cộng đồng, *clique*, đang được phát triển bởi xây dựng dựa trên các tam giác. Một số định nghĩa về cộng đồng trực tiếp dựa trên số lượng tam giác trong các đồ thị. Tuy nhiên, nó không mô tả cấu trúc của cộng đồng bằng những số tam giác có sẵn; các thông số khác cũng được thêm vào để phù hợp cho định nghĩa về cộng đồng. Cuối cùng, khi ta muốn phân hủy các đồ thị, kết nối không phải là lựa chọn duy nhất; tam giác cũng có thể mang lại một số kết quả tốt.

2.3.2. Thuật toán k-core địa phương

Ở phần trên ta định nghĩa các khái niệm về *k-core* địa phương và chứng

minh mối quan hệ của nó với *k-core* và xóa lân cận. Một *k-core* địa phương thu được bằng cách đệ quy loại bỏ tất cả các mức độ lân cận ít liên quan đến *k*, cho đến khi tất cả các đỉnh trong đồ thị còn lại có mức độ *k* ít tham gia lân cận cộng đồng. Ta phát triển các thuật toán 1 cho địa phương phân hủy *k-core*.

Trong thuật toán đầu vào là đồ thị *G* và *k*; đầu ra là các địa phương *k-core* đồ thị con. Trong bước 2, các chương trình đang chạy cho đến khi thỏa mãn điều kiện trong bước 10, có nghĩa là *k-core* địa phương đạt đến một điểm cố định. Kể từ *khi k-core* địa phương dựa trên đồ thị khu phố, trong bước 3 ta duyệt các đỉnh trong đồ thị một lần tại một thời gian; sau đó xây dựng mạng một khu phố *H*. Mỗi mức độ cộng đồng địa phương được tính bằng cách tìm các thành phần kết nối địa phương. Vì vậy trong bước 6 nếu thứ tự của mức độ cộng đồng địa phương $|E_C|$ nhỏ hơn *p*, tất cả các cạnh có các đỉnh *v* kết nối với mỗi đỉnh trong thành phần kết nối địa phương được đánh dấu. Trong bước 9 ta loại bỏ nhãn tất cả các cạnh. Các cạnh chính có thể được đánh dấu hai lần trở lên; ta không cần phải đếm số lượng cách đánh dấu và loại bỏ được dựa trên hàm *Boolean* của các cạnh (đánh dấu hoặc không). Khi đồ thị *G*' không thay đổi nữa, nó được đảm bảo rằng không có đỉnh nào không được thỏa mãn định nghĩa của *k-core* địa phương; điểm cố định đã đạt được; tất cả các đỉnh trong đồ thị *G'* thuộc về *k-core* địa phương.

Thuật toán 2.5: Phát hiện các lân cận *p-core* trong đồ thị

Đầu vào: Cho đồ thị *G (V,E), p.*

Đầu ra: $K_E(p) \subseteq G$, $K_E(p) \subseteq Q$ là một lân cận *p-core* của đồ thị con.

1) $G' \Leftarrow G$;

2) *repeat*

3) *for each* $v \in V(G')$

4) $H \Leftarrow N_{G'}(v)$

5) *for each* $E_j \in H$

6) *if* $|v_{Ej}| < p$

7) *then for* $u \in V(E_j)$

8) *do mark*(v, u)

9) $E(G') \Leftarrow \{(v,u) \in E(G') | (v,u)\}$ // không được đánh dấu

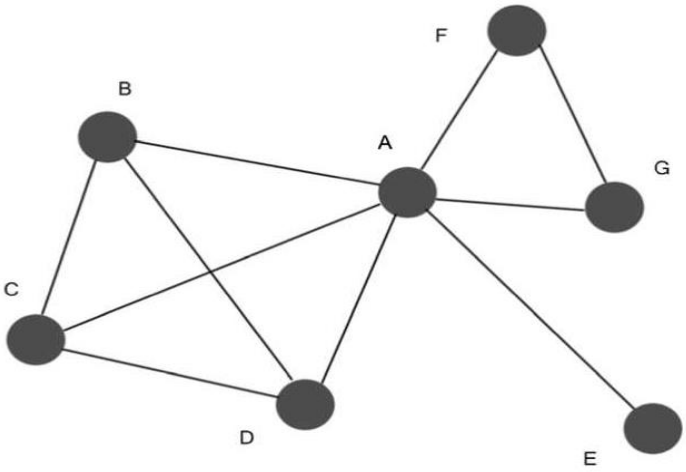
10) *until G' //*Không còn thay đổi

11) *return* $K_E(p) \Leftarrow G$

Trong hình 2.6 là một ví dụ về thuật toán. Khi ta cố gắng để tìm thấy những địa phương ở đồ thị con 3 core, ta bắt đầu duyệt với các đỉnh *A*: từ mạng các vùng lân cận, 3 thành phần kết nối địa phương được thu thập: {*B, C, D*} {*E*} và {*F, G*}; mức độ cộng đồng địa phương là có 3, 1 và 2. Bước tiếp theo chỉ có các thành phần {*B, C, D*} đáp ứng các yêu cầu.

Tất cả các cạnh đó các đỉnh *A* kết nối với {*E*} và {*F, G*} được đánh dấu. Sau đó, đỉnh *B* đang chạy cùng một quá trình và các đỉnh khác cũng như vậy. Cuối cùng chỉ có đỉnh thiết lập {*A, B, C, D*} là ở các địa phương *3-core* trong khi các đỉnh là *4 – core* nằm trong thuật toán *k-core*.

Định nghĩa của *k-core* địa phương đưa ra một lời giải thích rõ ràng về câu hỏi liệu các đồ thị *G'* là một địa phương *k-core* ở một điểm cố định ?. Mỗi đỉnh là duyệt đồ thị các vùng lân cận và tìm kiếm các thành phần kết nối địa phương mà không đáp ứng được các *k-core* địa phương.



Hình 2.7: Một ví dụ biểu đồ nhỏ cho việc tìm kiếm địa phương 3 – core từ thuật toán. {A, B, C, D} thuộc về địa phương 3 – core [10].

Yêu cầu: Các cạnh được đánh dấu trong các thành phần kết nối địa phương được gỡ bỏ không ảnh hưởng đến các kết nối của tập đỉnh trong thành phần kết nối địa phương ở chính nó. Từ điểm cố định trong thuật toán *k-core* địa phương sẽ có mỗi đỉnh trong đồ thị *G'* với ít nhất mức độ *k* cộng đồng lân cận.

Kết quả chứng minh được tính đúng đắn của thuật toán.

Các thuật toán k-core địa phương tập trung vào các nội dung của kết nối mà các thuật toán *k-core* thường bỏ qua. Như ta đã chứng minh trước đây, các thuật *toán k-core* địa phương bao gồm các thông tin về cấu trúc của cộng đồng. Mặc dù các thuật toán *k-core* địa phương không phải là một thuật toán phát hiện cộng đồng, nó cho thấy một số cấu trúc *clique* giống như trong biểu đồ, trong đó liên quan đến các cấu trúc mạng cộng đồng và lợi ích trong phân tích mạng. Từ định nghĩa của *k-core* địa phương, khi đồ thị lân cận chỉ có một thành phần kết nối, các thuật toán cũng giống nhau như các *k-core*. Với sự gia tăng của *k* trong *k-core* địa phương, các *k-core* địa phương trở nên dày đặc hơn và nhiều hơn khi chỉ có một thành phần kết nối; thuật toán *k-core* tại địa phương có thể được chuyển hóa thành thuật toán *k-core*. Nó sẽ xảy ra khi số *k* là tương đối lớn trong đồ thị.

Để đánh giá được mức độ liên kết hay liên thông giữa các cấu trúc cộng đồng trong mạng xã hội, các nhà nghiên cứu đã đưa ra một số thuật toán xử lý và phân tích các mối quan hệ giữa các liên kết cộng đồng với nhau. Như vậy ở chương 2 việc tìm hiểu và phân tích một số thuật toán nhanh tìm core là tiền để trong việc tìm hiểu và xây dựng bài toán ứng dụng phân tích mạng xã hội cho chương 3.

CHƯƠNG 3.
ỨNG DỤNG CỦA CORE TRONG PHÂN TÍCH MẠNG XÃ HỘI

Nội dung cơ bản trong chương này: Giải quyết một bài toán thực tế sử dụng lý thuyết đồ thị và thuật toán tìm core. Hiện thực hóa bằng chương trình ứng dụng.

3.1. Mô tả bài toán phân tích mạng mạng xã hội

Ở trong chương 1 ta có các khái niệm liên quan về mạng xã hội. Lịch sử các trang mạng xã hội ra đời trước các trang web truyền thống xã hội và hàng loạt các trang mạng xã hội tiếp theo...

Bài toán phân tích mạng xã hội lấy từ bảng số liệu thử nghiệm trên hệ thống một số mạng xã hội phổ biến ở Trung Quốc, và một số mạng xã hội phổ trên thế giới theo số liệu 2013 được trình bày trên cơ sở lý thuyết đồ thị (chương1), và xây dựng dựa trên thuật toán nhanh phân rã k-core địa phương (chương 2). Trong bài toán này tác giả tìm hiểu và lấy từ thực tế một số mạng xã hội sau đó phân tích để đánh giá mức độ liên thông giữa các mạng trong xã hội với nhau.

Trong phần này, kết quả và phân tích thí nghiệm được trình bày và thử nghiệm trên cấu hình máy tính Core i5 có tốc độ CPU 1,60*GHz*; DDR3 4*Gb*; chạy trên hệ điều hành Windows 8.1 Pro; các bộ dữ liệu có thể được tìm thấy trên bảng 3.1

Bảng 3.1: Lấy Cơ sở dữ liệu thử nghiệm; davg là mức độ trung bình của mạng; dmax là mức độ tối đa của mạng; r là sự phân cụm; c là hệ số cụm[11].

Mạng	Đỉnh	Cạnh	d_{avy}	d_{max}	r	C
FangYao	*383*	*3944*	*20.595*	*212*	*-0.1324*	*0.7467*
Net Science	*1589*	*2742*	*3.451*	*34*	*0.4616*	*0.6378*
Dolphin	*62*	*159*	*5.129*	*12*	*-0.0436*	*0.2590*
AS-JULY06	*22963*	*48436*	*4.2*	*2390*	*-0.1984*	*0.2304*
EMAIL-Enron	*36692*	*183831*	*10.02*	*1383*	*-0.1108*	*0.4970*
FOOTBALL	*115*	*613*	*10.66*	*11*	*0.1624*	*0.4032*
CA-ContMa	*23133*	*93497*	*8.08*	*280*	*0.1364*	*0.6336*
CA-AstroPh	*18772*	*198,110*	*21.10*	*504*	*0.2053*	*0.6308*
CA-GrQc	*5242*	*14496*	*5.53*	*81*	*0.6594*	*0.5302*
CA-HepTh	*9877*	*25998*	*5.26*	*65*	*0.2685*	*0.4717*

3.2. Phân tích mạng xã hội bằng thuật toán k-core địa phương

3.2.1. Đặt bài toán

Xuất phát từ thực tế khi tìm hiểu về các mạng xã hội, tác giả luận văn muốn áp dụng các nội dung kiến thức tìm hiểu được về lý thuyết đồ thị nói chung và khái niệm core nói riêng để giải quyết bài toán. Ngoài việc mô tả cài đặt thuật toán từ thực tế, đây thực sự cũng là một câu hỏi có giá trị. Chẳng hạn việc mô tả các lân cận giữa các mạng xã hội liên quan đến lý thuyết đồ thị: Các lân cận gần nhau nhất, mật độ trung bình, cây bao trùm. ....

Trong bảng dữ liệu 3.1 được tìm thấy trên trang web của SNAP [11], để đơn giản hóa, tất cả các tập dữ liệu được coi là đồ thị vô hướng hoặc đồ thị đối xứng có hướng. Các đồ thị sẽ không theo các vòng thứ tự.

Trong bảng 3.1, ta thấy có rất nhiều loại cơ sở dữ liệu mạng khác nhau: Mạng lưới cộng tác: trong các mạng này, các nút đại diện cho người hoặc đối tượng; các cạnh biểu thị mối quan hệ thông tin liên lạc (như mạng Football, Email-Enron) hoặc phối hợp (như CA-Hepth, NetScience). Mạng như vậy thường có một hệ số kết cụm so sánh cao.

Mạng chuyển hóa: các nút này là DNA hoặc chất chuyển hóa; các cạnh cho thấy chúng có một chức năng tích cực hoặc phản ứng hóa học với các tính chất chuyển hóa hoặc các loại phụ thuộc khác. Một số mạng là siêu đồ thị (như FangYao).

Mạng công nghệ: các nút là các router hoặc máy chủ. Các cạnh đại diện cho truyền thông hoặc kết nối vật lý giữa chúng.

3.2.2. So sánh giữa thuật toán địa phương với core và core lân cận

Trước tiên, ta so sánh thuật toán k–core lân cận và k-core dựa theo số đỉnh trong k-core có mức k khác nhau. Bảng 3.2 cho biết chỉ số k-core lân cận tối đa, $k_L^{\max}$ và chỉ số k tối đa, k^{max} và số lượng các đỉnh trong cả hai đồ thị k-core. Ta có $k_L^{\max}$ tương đương với k^{max}. Trong thực tế các thuật toán k- core lân cận đang dần loại bỏ các kết nối cục bộ của đồ thị (mức độ cộng đồng lân cận) mà vẫn thuộc (mức độ) liên kết toàn cầu. Với sự gia tăng của *k,* mỗi đỉnh sẽ kết nối với nhau trong đồ thị còn lại. Cuối cùng k-core lân cận và thuật toán k-core là tương đương. Một quan sát khác là số đỉnh trong đồ thị con k-core lân cận $|K_L(k_L^{\max})|$ và

đỉnh ở k-core đồ thị con $|K(k^{max})|$ không phải luôn giống nhau. Đối với việc cơ sở dữ liệu tồn tại trong mạng phổ biến hiện nay $|K_L(k_L^{\max})|$ và $|K(k^{max})|$ có khả năng xảy ra giống nhau.

Mặt khác $|K(k^{max})|$ và $|K_L(k_L^{\max})|$ có xu hướng khác nhau với những cơ sở dữ liệu vốn hay có mức độ lân cận đều trong khi các mức độ cộng đồng lân cận thì không được như vậy. Thông tin về k-core lân cận cho biết thêm về cấu trúc của đồ thị.

Hình 3.1 cho thấy so sánh về số đỉnh trong mỗi k-core như là một hàm về chỉ số trong hai thuật toán này. Đối với mạng FangYao toàn bộ hình dạng giống hệt nhau. Ta biết rằng k-core địa phương có liên hệ với các hình tam giác, đó là một tham số của hệ số phân nhóm. Mạng FangYao có hệ số phân nhóm cao (c = 0,7467) mà hầu hết các đỉnh kết nối với nhau. Trong điều kiện như vậy k-core và thuật toán k-core lân cận có cùng một kết quả.

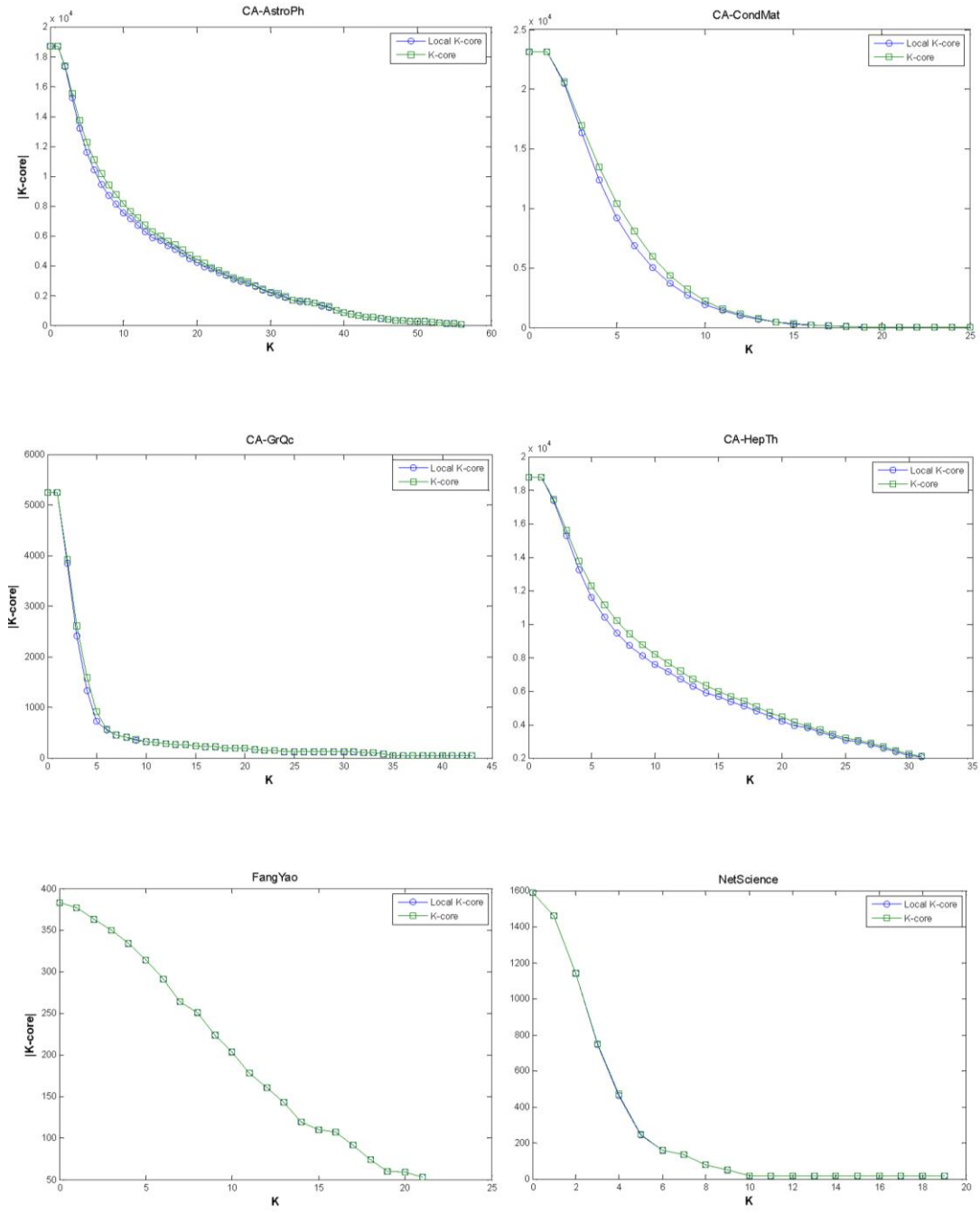
Bảng 3.2: So sánh với thuật toán k-core lân cận và k-core trong cơ sở dữ liệu; $k_L^{\max}$ là k-core số lân cận tối đa; kmax là số lượng tối đa của k-core; $|KL(k_L^{\max})|$ là số đỉnh của đồ thị con k-core lân cận khi k= $k_L^{\max}$; |K (kmax)| là số đỉnh của k-core đồ thị con khi k=kmax [11].

Mạng	$k_L^{\max}$	k^{max}	d_{max}	$\|K_L(k_L^{\max})\|$	$\|K(k^{max})\|$
FangYao	*2 2*	*22*	*212*	*53*	*53*
NetScience	*19*	*19*	*34*	*20*	*20*
Dolphin	*4*	*4*	*12*	*19*	*36*
AS-JULY06	*25*	*25*	*2390*	*71*	*71*
EMAIL-Enron	*43*	*43*	*1383*	*275*	*275*
FOOTBALL	*8*	*8*	*11*	*63*	*114*
CA-AstroPh	*56*	*56*	*504*	*57*	*57*
CA-ContMa	*25*	*25*	*280*	*26*	*26*
CA-GrQc	*43*	*43*	*81*	*44*	*44*
CA-HepTh	*31*	*31*	*65*	*32*	*32*

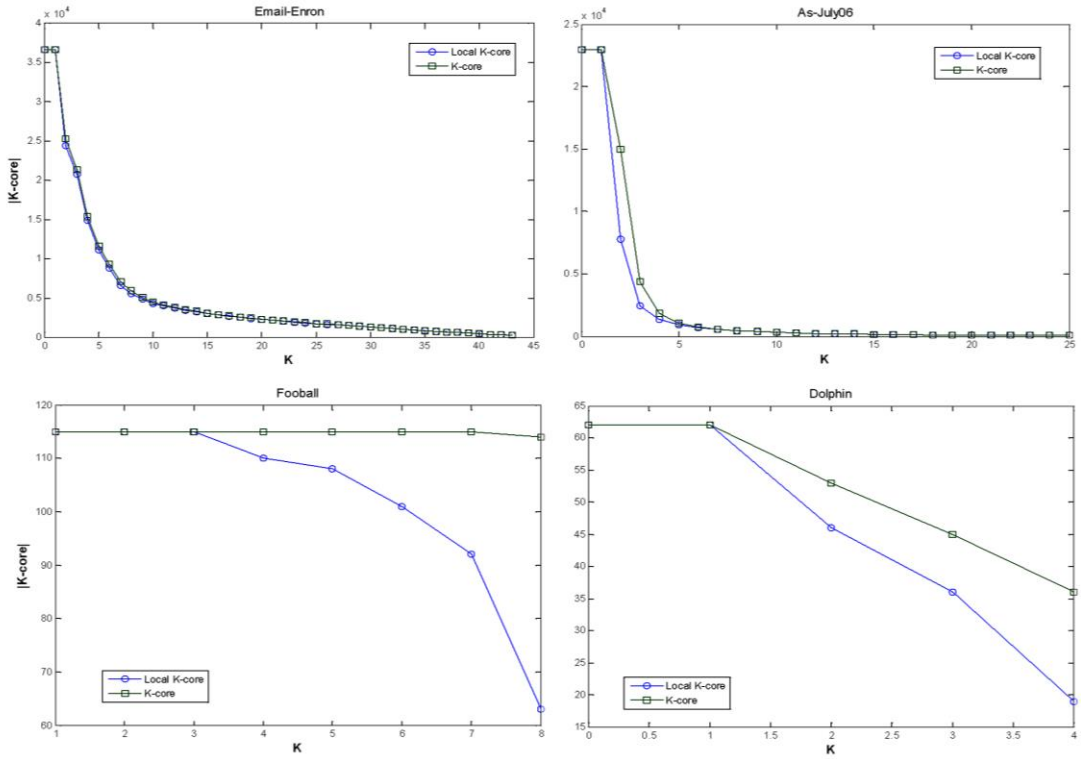
Kết quả:

Khi ta nhìn vào bốn bộ dữ liệu: CA-Astroph, CA-ContMa, CA-GrQc, CA-HepTh: Hình dạng đường cong cho k-core lân cận tất cả đều dưới một k-core. Điều đó có nghĩa từ mỗi bước đi, số đỉnh của k-core địa phương là nhỏ hơn so với một k-core. Ta nhận được sự quan sát tương tự ở hình 3.2. Qua sự quan sát để thấy sự phân biệt lớn trong các đường cong cho mạng Football và Dolphins.

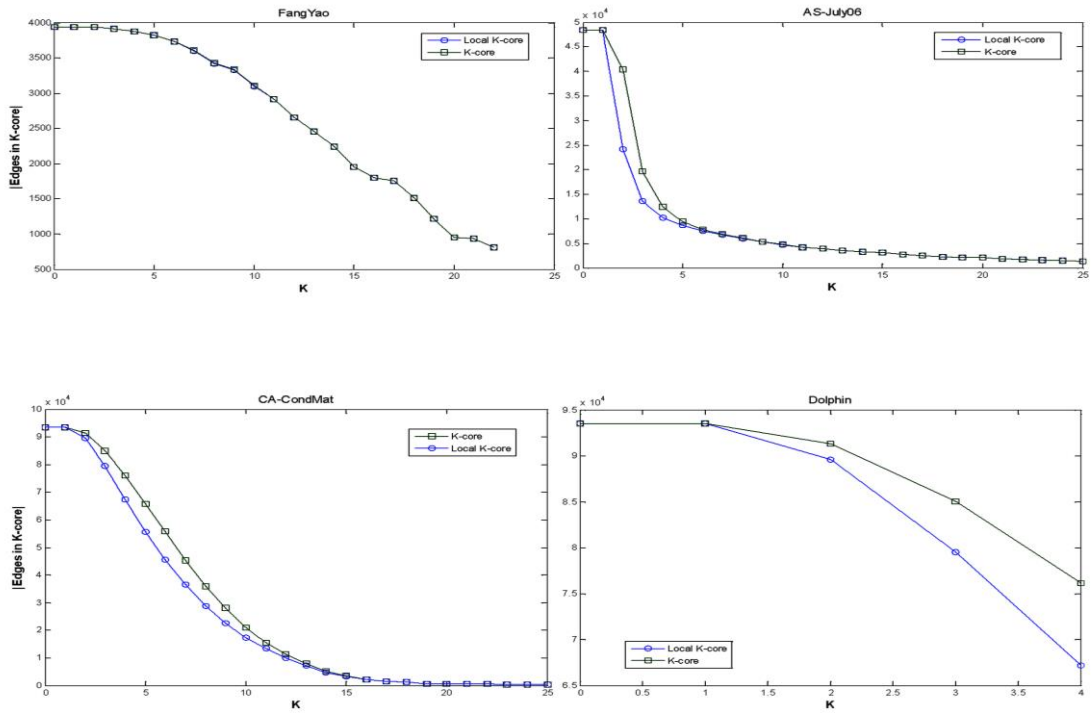
Hình 3.1 và 3.2 chứng minh được về k-core lân cận có ít đỉnh hơn khi so sánh với k–core trong cùng mức *k*. Hình 3.3 cho thấy việc so sánh các số cạnh của mỗi k-core như là một hàm các chỉ số có trong hai thuật toán này. Hình dạng đường cong cho mạng FangYao có tỷ lệ chồng chéo cao. Trong khi trên ba cơ sở dữ liệu khác, rất dễ thấy thuật toán k-core dài hơn.



Hình 3.1: Cơ sở dữ liệu số đỉnh của k –core như một hàm trong FangYao, NetScience, CA-AstroPh, CA-CondMat, CA-GrQc và CA-Hepth.



Hình 3.2: Cơ sở dữ liệu số đỉnh của k-core như một hàm trong Email-Enro, As-July06, Football và Dolphin.



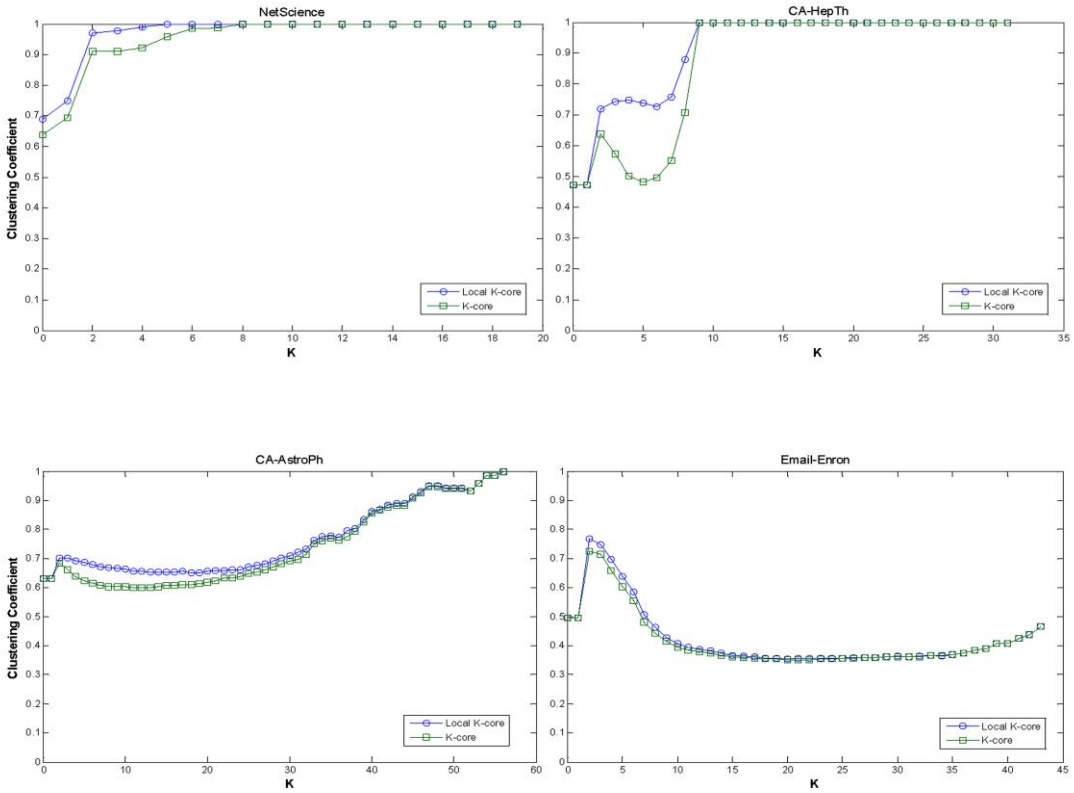
Hình 3.3: Cơ sở dữ liệu số cạnh của k-core như một hàm trong FangYao, As-July06, CA-CondMat và Dolphins.

Thuật toán cho thấy số cạnh k-core ít hơn số điểm lân cận điều này đúng với việc phân tích ở các đồ thị trước đấy.

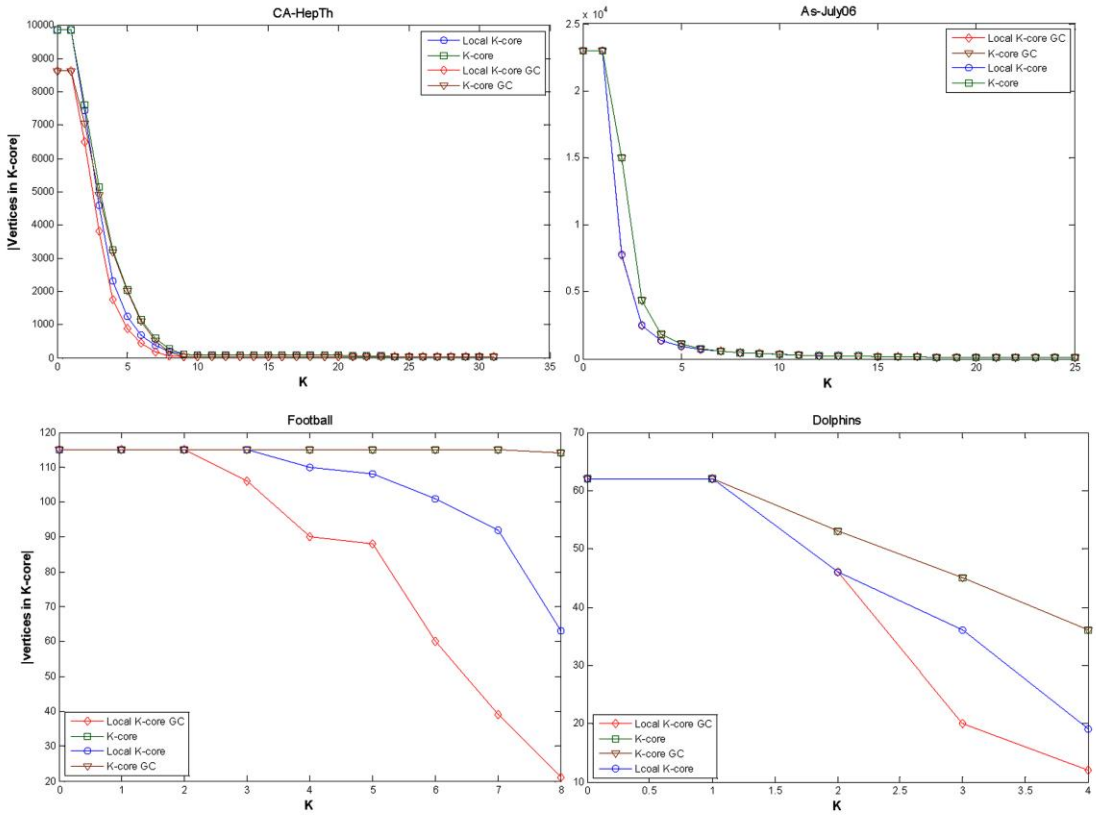
3.3. So sánh hệ số phân nhóm trong thuật toán k-core

Ở phần này ta đi chứng minh mối quan hệ k-core lân cận với sự xen kẽ 3-clique. Hình 3.4 cho thấy sự biến đổi của hệ số phân lớp k–core với các *k* khác nhau trong cả hai thuật toán. Thuật toán k–core lân cận luôn có một hệ số phân cụm cao hơn ở cùng một mức *k* so với thuật toán k–core. Tức là thuật toán k-core lân cận giữ cho thông tin cấu trúc rất tốt khi so sánh với k-core. Ở đây k-core lân cận chắc chắn là một k-core trong khi điều ngược lại là không đúng. Một quan sát khác khi *k = 2*, k-core lân cận thường có một bước nhảy lớn trong đường cong. Như đã đề cập ở trên, khi *k = 2*, tất cả các cấu trúc không phải là tam giác được loại bỏ vì hệ số phân cụm có liên quan đến số hình tam giác trên đồ thị.

Hệ số phân cụm tăng bằng cách loại bỏ các cấu trúc hình tam giác. Ngoài ra, khi *k = 56* tại CA-Astroph, *k = 6* trong NetScience và *k = 9* trong mạng CA-HepTh, hệ số phân lớp cuối cùng trở thành *1*, có nghĩa là đồ thị con k-core trở thành một tập hợp các nhóm. Khi đồ thị con k-core trở thành liên quan, thường không cần phải phân hủy thêm nữa vì đây là điều kiện cân bằng cho biểu đồ. Rõ ràng là k-core địa phương sớm đạt được điều kiện cân bằng hơn k-core.



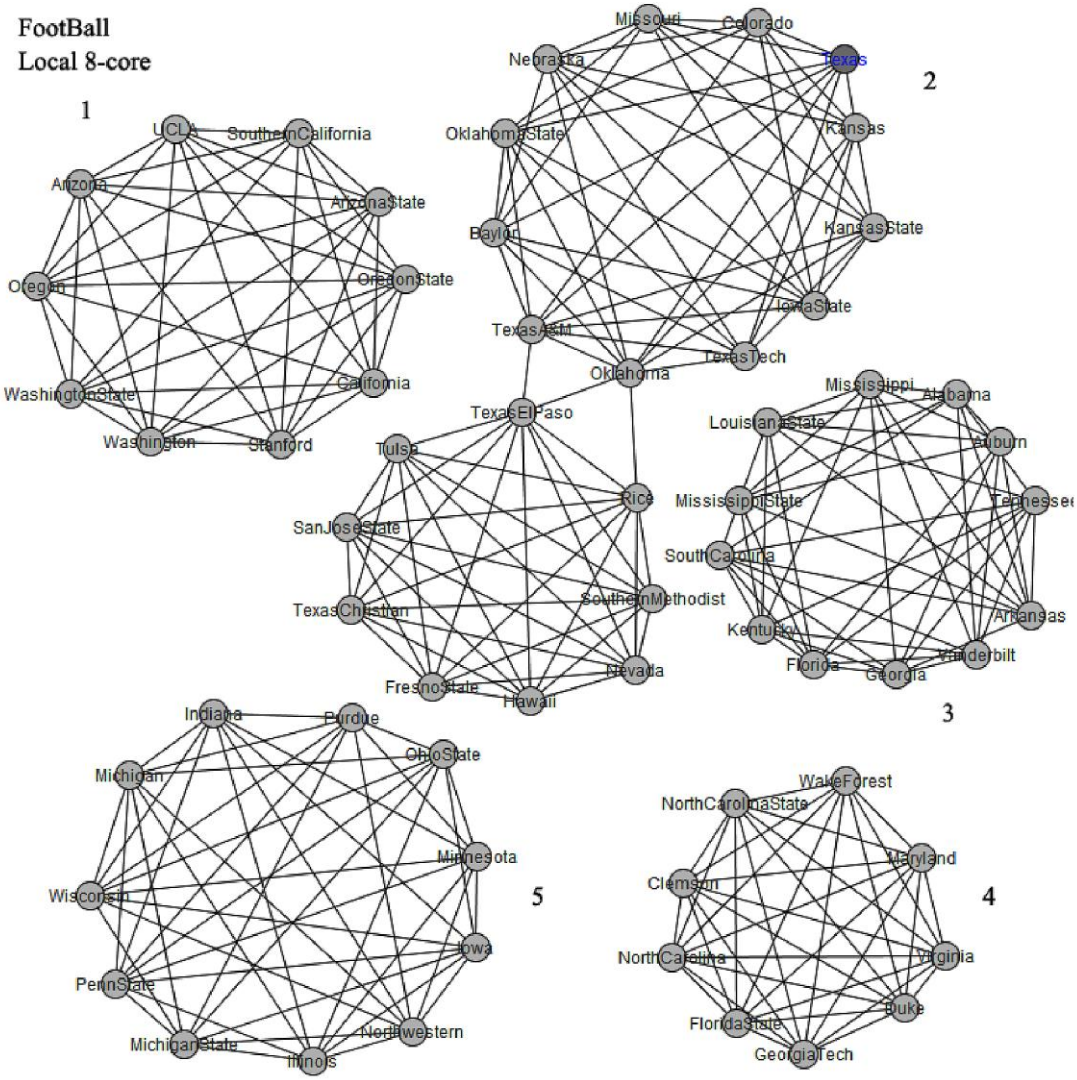
Hình 3.4: Cơ sở dữ liệu thu gọn hệ số của k –core như là một chức năng trong CA-AstroPh, Email-Enron, NetScience và CA-HepTh.



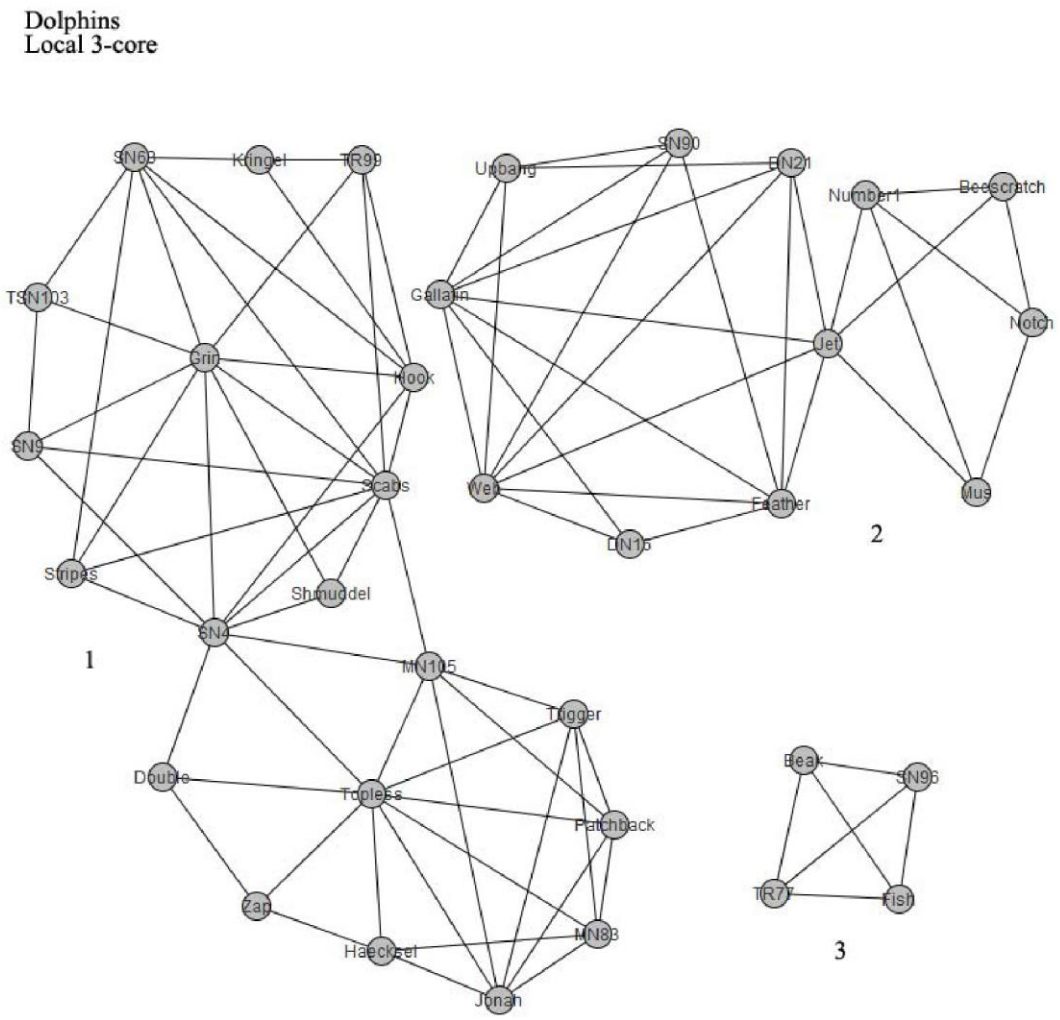
Hình 3.5: Cơ sở dữ liệu kích thước các thành phần khổng lồ và kích thước của k-core như là một chức năng trong CA-HepTh, As-July06, Football và Dolphins

Cấu trúc k-core cộng đồng tại địa phương.

Từ hai thí nghiệm trên, cho thấy thuật toán k-core tại địa phương có yêu cầu nghiêm ngặt hơn k-core. Trong cơ sở dữ liệu ta tập trung vào đồ thị phụ k-core. Hình 4-5 cho biết sự đa dạng về kích thước thành phần hai thuật toán ở mức độ *k* khác nhau. Ở đây các thành phần có kết nối là lớn nhất.



Hình 3.6: 8-core lân cận trong mạng lưới Footboall ở 63 đỉnh hợp thành 21 đỉnh. Biểu đồ được hiển thị bởi Java Jung package [12].



Hình 3.7: 3-core lân cận core trong mạng lưới Dolphins ở 36 đỉnh hợp thành 20 đỉnh. Biểu đồ được hiển thị bởi gói Java Jung package [12].

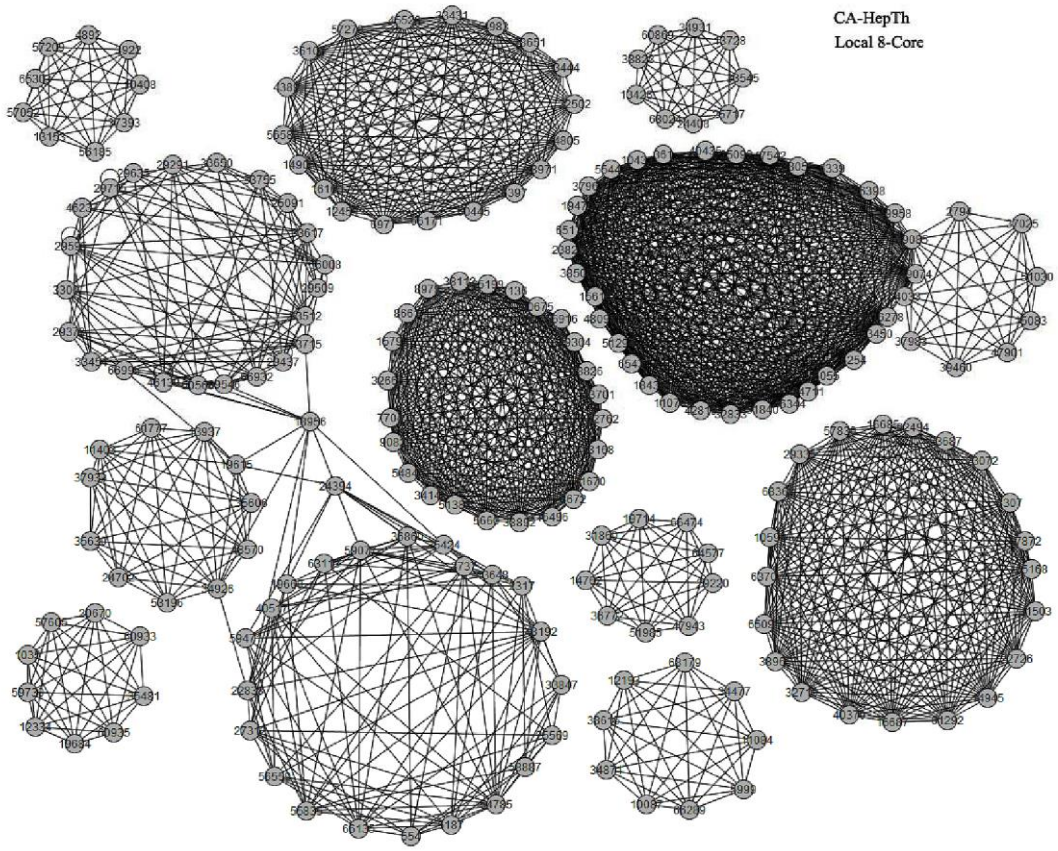
Liên kết trong biểu đồ hình 3.5 ta nhận thấy rằng, đối với thuật toán k-core các thành phần lớn gần như cùng kích cỡ với đồ thị k-core, có nghĩa k-core không thể hiện tốt cấu trúc cộng đồng của đồ thị. Đối với Thuật toán k-core tại địa phương, đặc biệt là trong mạng CA-HepTh, Football và Dolphins các thành phần lớn nhỏ hơn nhiều so với k-core, cho phép ta có thể xem cấu trúc đồ thị của cộng đồng.

Hình 3.6 hiển thị mạng nội bộ 5-core của mạng lưới bóng đá trong đó có 5 thành phần được kết nối; Mỗi thành phần kết nối là một cấu trúc giống như clique. Rõ ràng ngay cả trong Giant Component (đánh dấu bằng 2) cũng có thể dễ dàng chia thành 2 thành phần kết nối, nếu ta sử dụng một thuật toán phân cụm

đồ thị cơ bản. Các địa phương 5-core có 63 đỉnh và kích thước thành phần cụm lớn là 21; Trong khi 5–core có 114 đỉnh và kích thước thành phần cụm lớn là 114.

Hình 3.6 cho thấy các đỉnh có kết nối tương tự nên nằm trong cùng 1 cộng đồng; khi so sánh với kết quả phân loại thực của mạng football, giả sử các đỉnh trong 1 hợp phần thuộc phân cụm PAC, các đỉnh trong Component 3 thuộ SEC, 4 thuộc ACC, và 5 thuộc B10. Cũng trong 2 hợp phần, các đỉnh thuộc phân cụm B12 và C-USA.

Các hiện tượng tương tự được trình bày trong hình 3.7 cho mạng lưới Dolphins và hình 3.8 cho mạng CA-HepTh. Đồ thị con k-core luôn có số đỉnh lớn hơn và kích thước thành phần khổng lồ so với k core địa phương ở cùng mức k: 45 đỉnh và cỡ thành phần khổng lồ 45 so với 36 đỉnh và 20 khi $k = 2$ trong mạng Dolphins; 255 đỉnh và cỡ thành phần khổng lồ 172 so với 206 đỉnh và 57 khi $k = 5$ trong mạng CA-HepTh. Ở đây thuật toán k core địa phương hiển thị cấu trúc cộng đồng trong k-core mà không bao giờ hiển thị trong thuật toán k-core bình thường.



Hình 3.8: 8-Core lân cận trong mạng CA-HepTh ở 206 đỉnh hợp cụm 57 đỉnh lớn. Biểu đồ được hiển thị bởi gói Java Jung package [12].

Trong chương 3 này, đồ thị khu phố có xu hướng là một cái nhìn hay về phát hiện cấu trúc cộng đồng được biểu diễn bằng biểu đồ; Khái niệm về mức độ cộng đồng địa phương được xác định bởi kích thước thành phần được kết nối trong biểu đồ khu phố. Sự kết nối xa hơn giữa *k-core* lân cận, đảm bảo ít nhất mức độ cộng đồng địa phương *k* trong đồ thị phụ, góc nhìn của đồ thị, và sự xen kẽ của Clique, giống cấu trúc cộng đồng như clique với mật độ cao. Ngoài ra, thuật toán *k-core* lân cận nếu mỗi đồ thị lân cận chỉ có một thành phần kết nối cần mật độ cao của đồ thị. *k –core* địa phương là một ý tưởng mở rộng của k-core với điều kiện hạn chế.

Thông qua thử nghiệm, chứng minh rằng thuật toán k-core tại địa phương có thể có ít cạnh và đỉnh hơn thuật toán k-core, trong khi giúp đồ thị có mật độ cao hơn. Đồng thời, thuật toán k-core địa phương cho thấy một cấu trúc cộng

đồng tốt trong quá trình phân rã k-core, đây là một cách tốt hơn để xem cấu trúc đồ thị.

K-core tại địa phương có một mối quan hệ với sự xen kẽ 3-clique. Có cấu trúc cộng đồng mới khác như k-core khu vực lân cận đỉnh thỏa mãn điều kiện sự xen kẽ 4-clique và điều kiện đó cũng liên quan đến đồ thị k kết nối.

Tóm lại, với mạng phức tạp, thuật toán tìm core cũng như một số thuật toán trong lý thuyết đồ thị đã cho thấy tính hiệu quả của và giúp ta hiểu sâu sắc hơn, rút ra nhiều ý nghĩa từ tập dữ liệu mạng phức tạp.

KẾT LUẬN

Dưới sự hướng dẫn của Giáo viên hướng dẫn – TS. Trương Hà Hải cùng với sự nỗ lực của bản thân, luận văn đã đạt được một số kết quả sau:

1. Tìm hiểu các kiến thức cơ bản về Lý thuyết đồ thị, mạng xã hội.

2. Học hỏi và nắm bắt được một số thuật toán xử lý trong đồ thị, đặc biệt là các thuật toán phân rã đồ thị, tìm k-core, p-core với thời gian đa thức.

3. Nghiên cứu cài đặt được các thuật toán đã tìm hiểu, phát biểu và giải quyết được một số bài toán thực tế áp dụng những kiến thức đã nghiên cứu.

4. Các lý thuyết và thí nghiệm trình bày trong chương 2 và chương 3 ở đây vẫn còn sơ bộ và cần phải được mở rộng bằng các nghiên cứu thực hiện trong tương lai. Thứ nhất, lý thuyết và các thí nghiệm về k-core địa phương chỉ tập trung vào dữ liệu mạng thực; đối với nghiên cứu, chẳng hạn như mạng ngẫu nhiên hoặc giới hạn thấp cho sự xuất hiện của k-core địa phương, không được đề cập trong luận án. Tốt nhất là nên xây dựng cơ sở mạng k-core cục bộ ngẫu nhiên trên các tham số như k và hệ số phân cụm. Một lĩnh vực khác là, thay vì tập trung vào mức độ cộng đồng địa phương (kích cỡ của các thành phần kết nối trong đồ thị lân cận), số lượng các thành phần được kết nối có thể là một cuộc tranh luận cho các nghiên cứu sâu hơn; k-core cũng liên quan đến cấu trúc cộng đồng và có thể là một phương pháp lân cận tốt cho đồ thị. Nếu một đồ thị lân cận của đỉnh có nhiều thành phần kết nối, nó có thể có nghĩa là đỉnh có thể là một điểm nóng, thay vì một thành viên của cộng đồng.

Trong thời gian tới luận văn có thể làm nền tảng để phát triển thêm: Nghiên cứu cải tiến các thuật toán cho tốt hơn; Tìm tòi các bài toán thực tế để vận dụng giải quyết mang lại ý nghĩa thực tế.

Mặc dù đã rất nỗ lực, nhưng thời gian và kiến thức hạn chế nên chắc chắn luận văn còn nhiều thiếu sót và kết quả đạt được chưa nhiều. Kính mong các Thầy Cô góp ý để luận văn được hoàn thiện hơn.

Trân trọng cảm ơn.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt

[1] Nguyễn Cam, Chu Đức Khánh, *Lý thuyết đồ thị*, NXB Thành phố Hồ Chí Minh, 1999

[2] Trần Minh Đức, *Nghiên cứu, thiết kế, thử nghiệm mạng xã hội phục vụ phát triển nông thôn*, Học viện Công nghệ bưu chính viễn thông Hà Nội, 2012

Tài liệu tiếng Anh

[3] Kayhan Erciyes, *Complex Networks: An Algorithmic Perspective* Sep 6, 2014

[4] Niklaus Wirth, *Algorithms + Data Structures = Programs (Prentice-Hall Series in Automatic Computation)* Feb 1976

[5] Nasrullah Memon and Reda Alhajj (Editor), *From Sociology to Computing in Social Networks: Theory, Foundations and Applications (Lecture Notes in Social Networks)* 2010 th Edition

[6] *VladimirBatagelj, and Matjaz˘ Zaversnik˘,* Cores Decomposition of Networks, *UniversityofLjubljana,Slovenia, September 24–27, 2001*

[7] Vladimir Batagelj and Matjaz Zaversnik, *An O(m) Algorithm for Cores Decomposition of Networks*, University of Ljubljana, 25 Oct 2003

[8] Vladimir Batagelj, and Matjaz Zaversnik, *Fast algorithms for determining (generalized) core groups in social networks*, 5 November 2011

[9] Online, Mon, 03 Jan 2011, *Social Network Analysis Theory and Applications*

[10] Chen Lu, Local K-Core Algorithm in Complex Networks , Master of Compute Science, Harbin University of Science and Technology, Harbin, ChiNa, 2002

[11] Jure Leskovec. (2, May, 2013). *SNAP*. http://snap.stanford.edu/data/index.html

[12] The JUNG Framework Development Team. (24, May, 2013). *JUNG*. http://jung.sourceforge.net/index.html