Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jsi.2017.021

KHAI THÁC TẬP PHỔ BIẾN TỪ DỮ LIỆU GIAO DỊCH VỚI NHIỀU NGƯỠNG PHỔ BIẾN TỐI THIỂU TRÊN BỘ XỬ LÝ ĐA NHÂN

Phan Thành Huấn[1] và Lê Hoài Bắc[2]

[1]Bộ môn Tin học, Trường Đại học Khoa học Xã hội và Nhân văn, ĐHQG-HCM
[2]Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

Thông tin chung:
Ngày nhận bài: 15/09/2017
Ngày nhận bài sửa: 10/10/2017
Ngày duyệt đăng: 20/10/2017

Title:
Mining frequent itemsets in transactional databases with multiple minimum support threshold on multiple-core processors

Từ khóa:
Bộ xử lý đa nhân, luật kết hợp, nhiều ngưỡng phổ biến tối thiểu, tập phổ biến, thuật toán song song

Keywords:
Association rule mining, frequent itemsets, multiple-core processor, multiple minimum support thresholds, parallel algorithm

ABSTRACT

Association rule mining, one of the most important and well-researched techniques of data mining. Mining frequent itemsets are one of the most fundamental problems and most time-consuming in association rule mining. Most of the algorithms in literature used to find frequent itemsets satisfying single minimum support threshold. In practice, frequentcy of each item reflects the nature and role of items in transactional databases. This paper proposes an efficient mining parallel algorithm for frequent itemsets with multiple minimum support thresholds (a different minimum item support for each item) on Multiple-core Processors. Proposed algorithm easily extends on distributed computing systems as Hadoop, Spark. Finally, result experiments presented on both synthetic and real-life datasets show the better proposed algorithm than the existing algorithms.

TÓM TẮT

Trong khai thác dữ liệu, kỹ thuật quan trọng và được nghiên cứu nhiều là khai thác luật kết hợp. Khai thác tập phổ biến là một trong những bước cơ bản và chiếm nhiều thời gian trong khai thác luật kết hợp. Hầu hết các thuật toán tìm tập phổ biến thỏa một ngưỡng phổ biến tối thiểu duy nhất. Trong thực tế, độ phổ biến của từng mục hàng phản ánh bản chất, vai trò của mục hàng trong các giao dịch. Trong bài viết này, chúng tôi đề xuất thuật toán song song khai thác hiệu quả tập phổ biến với nhiều ngưỡng phổ biến tối thiểu (mỗi mục hàng có một ngưỡng phổ biến tối thiểu riêng) trên bộ xử lý đa nhân. Thuật toán đề xuất dễ dàng mở rộng trên nhiều hệ thống tính toán phân tán như Hadoop, Spark. Sau cùng, chúng tôi trình bày kết quả thực nghiệm trên bộ dữ liệu thực và giả lập cho thấy thuật toán đề xuất hiệu quả hơn so với thuật toán hiện hành.

Trích dẫn: Phan Thành Huấn và Lê Hoài Bắc, 2017. Khai thác tập phổ biến từ dữ liệu giao dịch với nhiều ngưỡng phổ biến tối thiểu trên bộ xử lý đa nhân. Tạp chí Khoa học Trường Đại học Cần Thơ. Số chuyên đề: Công nghệ thông tin: 155-163.

1 GIỚI THIỆU

Khai thác luật kết hợp là một kỹ thuật quan trọng trong lĩnh vực khai thác dữ liệu. Mục tiêu khai thác là phát hiện những mối liên hệ giữa các giá trị dữ liệu trong dữ liệu giao dịch. Mô hình đầu tiên của bài toán khai thác luật kết hợp là mô hình nhị phân hay còn gọi là mô hình cơ bản (Agrawal *et al.*, 1993), phân tích dữ liệu giao dịch, phát hiện các mối liên hệ giữa các tập mục hàng hoá đã bán được tại

các siêu thị. Từ đó, doanh nghiệp có kế hoạch bố trí, sắp xếp, kinh doanh hợp lý, đồng thời tổ chức sắp xếp các quầy gần nhau để có doanh thu trong các phiên giao dịch là lớn nhất.

Bài toán khai thác luật kết hợp là khai phá các luật kết hợp có độ phổ biến (*support*) cũng như độ tin cậy (*confidence*) lớn hơn hoặc bằng một ngưỡng phổ biến tối thiểu (*minsup*) và ngưỡng tin cậy tối thiểu (*minconf*).

Các thuật toán được đề xuất để khai thác luật kết hợp chia thành 2 giai đoạn (Agrawal *et al.*, 1993, 1994; Han *et al.*, 2004):

Giai đoạn 1: Tìm tất cả các tập mục phổ biến từ dữ liệu giao dịch thoả *minsup*;

Giai đoạn 2: Sinh các luật tin cậy kết hợp từ tập mục phổ biến tìm thấy ở giai đoạn thứ nhất.

Giai đoạn *thứ nhất* chiếm hầu hết thời gian cho quá trình khai thác luật kết hợp. Giá trị ngưỡng phổ biến tối thiểu *minsup* là yếu tố quan trọng trong quá trình rút gọn không gian tìm kiếm cũng như giới hạn các luật sinh trong giai đoạn *thứ hai*. Các thuật toán khai thác luật kết hợp truyền thống chỉ dùng một giá trị ngưỡng phổ biến tối thiểu *minsup* với ngầm định là các mục hàng có cùng tính chất và tần số trong dữ liệu, điều này không thực tế. Trong kinh doanh bán lẻ, thông thường các mặt hàng thiết yếu, hàng tiêu dùng và các sản phẩm giá rẻ được mua nhiều hơn, trong khi các mặt hàng xa xỉ và các sản phẩm giá trị cao lại ít được mua. Nếu chọn *minsup* quá cao thì các mặt hàng được khai thác thông thường có giá thành thấp và mang lại lợi nhuận không cao cho doanh nghiệp. Ngược lại, nếu chọn *minsup* quá thấp thì các mặt hàng được khai thác quá lớn, điều này làm cho doanh nghiệp khó khăn khi ra quyết định kinh doanh. Vì vậy, Liu *et al.* (1999) đã mở rộng bài toán khai thác luật kết hợp với nhiều ngưỡng phổ biến tối thiểu (*mỗi mục hàng có một ngưỡng phổ biến tối thiểu riêng*) tương ứng mỗi mục hàng khác nhau có tính chất khác nhau và tần số giao dịch khác nhau. Nhóm tác giả này đã đề xuất thuật toán **MSApriori** – khai thác luật kết hợp khác nhau thỏa ngưỡng phổ biến tối thiểu khác nhau phụ thuộc vào các mục hàng có trong luật.

Một số thuật toán điển hình khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu:

Thuật toán MSApriori (Liu *et al.,* 1999) được đề xuất để khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu khác nhau. Thuật toán này sử dụng phương pháp tiếp cận tựa **Apriori** và tính chất bao đóng được sắp xếp theo mục hàng để giảm không gian tìm kiếm, chi phí tính toán.

Thuật toán CFP-growth (Hu *et al*., 2006) đã đề xuất hướng tiếp cận tựa thuật toán **FP-growth** trong khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu được gọi là **CFP-growth**. Thuật toán sử dụng cấu trúc **MIS-tree** tựa **FP-tree** (Han *et al.*, 2004) để xuất để lưu trữ thông tin quan trọng các mẫu phổ biến. Thuật toán khai thác đầy đủ tập phổ biến với một lần quét dữ liệu. Thuật toán này tương đối tốt hơn so với **MSApriori**.

Thuật toán CFP-growth++ (Kiran *et al.*, 2011) đã đề xuất cải tiến thuật toán **CFP-growth** bằng cách rút gọn không gian tìm kiếm và xây dựng *MIS-tree nhỏ gọn* dựa trên *MIS-tree*. Thuật toán đề xuất bốn kỹ thuật rút gọn không gian tìm kiếm: *ngưỡng phổ biến tối thiểu thấp nhất*, *ngưỡng phổ biến tối thiểu có điều kiện*, *tính chất bao đóng có điều kiện và tỉa các nút lá không phổ biến*. Thuật toán cải thiện hiệu suất đáng kể so với thuật toán **CFP-growth**.

Các thuật toán trên chưa đáp ứng thực tế, khi cần khai thác luật kết hợp thì người dùng có thể yêu cầu thực hiện khai thác luật kết hợp thỏa nhiều ngưỡng phổ biến tối thiểu trong nhiều chuỗi thao tác liên tiếp khác nhau (xây dựng lại *MIS-tree* và rút gọn không gian từ đầu). Để đáp ứng thực tế, nhóm tác giả đề xuất thuật toán tuần tự **SEQ-MMSFI** – theo cấu trúc *2 Pha* và tái sử dụng *Pha 1* cho chuỗi thao tác tiếp theo. Từ đó, xây dựng thuật toán song song **MCP-MMSFI** khai thác nhanh tập phổ biến từ mảng chứa các *itemset* đồng xuất hiện và không đọc lại dữ liệu cho lần khai thác tiếp theo, bao gồm các thuật toán con sau:

– Xây dựng mảng **Index_COOC** chứa *itemset* đồng xuất hiện, *itemset* xuất hiện ít nhất trong một giao dịch của từng *item* hạt nhân;

– Thuật toán tuần tự **SEQ-MMSFI** khai thác hiệu quả tập phổ biến với nhiều ngưỡng phổ biến tối thiểu dựa trên mảng **Index_COOC**.

– Thuật toán song song **MCP-MMSFI** khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu trên bộ xử lý đa nhân (BXLĐN).

Trong phần 2, bài báo trình bày các khái niệm cơ bản về khai thác tập phổ biến truyền thống, tập phổ biến với nhiều ngưỡng phổ biến tối thiểu. Phần 3, xây dựng thuật toán xác định mảng chứa *itemset* đồng xuất hiện và *itemset* xuất hiện *ít nhất trong một giao dịch* của từng *item* hạt nhân và thuật toán tuần tự **SEQ-MMSFI** khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu. Phần 4, xây dựng thuật toán song song **MCP-MMSFI** khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu trên BXLĐN. Kết quả thực nghiệm được trình bày trong phần 5 và kết luận ở phần 6.

2 CÁC KHÁI NIỆM CƠ BẢN

2.1 Khai thác tập phổ biến truyền thống

Khai thác tập phổ biến truyền thống là các thuật toán (Agrawal *et al.*, 1993, 1994; Han *et al.*, 2004) dùng duy nhất *một* giá trị ngưỡng phổ biến tối thiểu *minsup* với ngầm định là các mục hàng có cùng tính chất và độ phổ biến trong dữ liệu. Các hạn chế khi khai thác tập phổ biến truyền thống: giá trị *minsup* cao thì các tập mục hiếm bị bỏ qua hoặc khi giá trị *minsup* thấp thì sinh tập mục phổ biến quá lớn. Sau đây là các khái niệm liên quan:

Cho $I = \{i_1, i_2, ..., i_m\}$ là tập gồm m mục hàng riêng biệt, mỗi mục hàng gọi là *item*. Tập các mục $X = \{i_1, i_2, ..., i_k\}, \forall i_j \in I (1 \le j \le k)$ gọi là *itemset*, tập mục có k mục gọi là *k-itemset*. $\mathcal{D}$ là dữ liệu giao dịch, gồm n bản ghi phân biệt gọi là tập các giao dịch $T = \{t_1, t_2, ..., t_n\}$, mỗi giao dịch $t_i = \{i_{k_1}, i_{k_2}, ..., i_{k_j}\}, i_{k_j} \in I (1 \le k_j \le m)$.

Định nghĩa 1: *Độ phổ biến* (support) của *itemset* $X \subseteq I$, ký hiệu *sup(X)*, là số các giao dịch trong $\mathcal{D}$ có chứa X.

Định nghĩa 2: Cho $X \subseteq I$, X gọi là *itemset* phổ biến nếu $sup(X) \ge minsup$, trong đó *minsup* là ngưỡng phổ biến tối thiểu. Ký hiệu **FI** là tập hợp các tập mục phổ biến.

Tính chất 1: $\forall X \subseteq Y$: $sup(Y) \ge minsup \Rightarrow sup(X) \ge minsup$;

Tính chất 2: $\forall X \subset Y$: $sup(X) < minsup \Rightarrow sup(Y) < minsup$;

Cho dữ liệu giao dịch $\mathcal{D}$ trong Bảng 1.

Bảng 1 : Dữ liệu giao dịch Ɗ

Mã giao dịch	Tập item
t1	A C E F
t2	A C G
t3	E H
t4	A C D F G
t5	A C E G
t6	E
t7	A B C E
t8	A C D
t9	A B C E G
t10	A C E F G

Ví dụ 1: Dữ liệu giao dịch $\mathcal{D}$ trong *Bảng 1*, có 8 *item* riêng biệt $I = \{A, B, C, D, E, F, G, H\}$ và 10 giao dịch $T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10\}$ với giá trị ngưỡng *minsup* = 2, ta có:

Tập mục $X = \{A, C, E\}$, $sup(ACE) = 5 \ge minsup$, ta nói: "$X = \{ACE\}$ phổ biến theo ngưỡng *minsup* = 2";

Theo ***tính chất 1*** thì các tập con của $X = \{ACE\}$ cũng phổ biến: các tập con của X đều phổ biến – $sup(A) = 8$, $sup(C) = 8$, $sup(E) = 7$, $sup(AC)$, $sup(AE) = 5$, $sup(CE) = 5 \ge minsup$.

Tương tự, với $Y = \{H\}$ thì $sup(H) = 1 < minsup$, ta nói: "$Y = \{H\}$ *không phổ biến* theo ngưỡng *minsup* = 2";

Theo ***tính chất 2*** thì các tập cha của $Y = \{H\}$ cũng *không phổ biến*, nghĩa là $Y = \{EH\}$ cũng *không phổ biến*, với $sup(EH) = 1 < minsup = 2$.

2.2 Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu

Trong thực tế, hầu hết dữ liệu giao dịch đều không đồng nhất về tính chất của từng mục hàng, cũng như tần số giao dịch của các mục hàng. Các tác giả đã đề xuất thuật toán khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu của từng mục hàng. Các thuật toán này khai thác luật kết hợp khác nhau thỏa ngưỡng phổ biến tối thiểu khác nhau phụ thuộc vào ngưỡng phổ biến của các mục hàng có trong luật. Sau đây là các khái niệm liên quan:

Cho $I = \{i_1, i_2, ..., i_m\}$ là tập gồm m mục hàng riêng biệt, mỗi mục hàng gọi là *item*. Tập $MIS = \{mis_{i_1}, mis_{i_2}, ..., mis_{i_m}\}$ là tập các ngưỡng phổ biến tối thiểu cho từng *item*. $\mathcal{D}$ là dữ liệu giao dịch, gồm n bản ghi phân biệt gọi là tập các giao dịch $T = \{t_1, t_2, ..., t_n\}$, mỗi giao dịch $t_i = \{i_{k_1}, i_{k_2}, ..., i_{k_j}\}, i_{k_j} \in I (1 \le k_j \le m)$.

Định nghĩa 3: Cho $X = \{i_1, i_2, \dots, i_k\}, \forall i_j \in I (1 \le j \le k)$, ngưỡng phổ biến tối thiểu của itemset X được tính là $mis_X = \min(mis_{i_1}, \dots, mis_{i_k}), \forall i_{j=\overline{1,k}} \in X$.

Định nghĩa 4: Cho $X = \{i_1, i_2, \dots, i_k\}, \forall i_j \in I (1 \le j \le k)$, X gọi là *itemset* phổ biến nếu $sup(X) \ge mis_X$.

Bảng 2: Ngưỡng phổ biến tối thiểu của từng mục hàng trong dữ liệu giao dịch Ɗ

Mục hàng	A	B	C	D	E	F	G	H
mis_{i_j}	5	2	3	3	2	4	3	2

Ví dụ 2: Dữ liệu giao dịch $\mathcal{D}$ trong *Bảng 1*, và mỗi mục hàng có mỗi giá trị ngưỡng phổ biến tối thiểu được cho trong Bảng 2, ta có:

Tập mục $X = \{A, C, E\}$, $sup(ACE) = 5 \ge mis_X = \min(mis_A, mis_C, mis_E) = min(5, 3, 2) = 2$, ta nói: "$X = \{A, C, E\}$ là tập mục phổ biến";

Tập mục $Y = \{A, C, F\}$, $sup(ACF) = 3 \ge mis_Y = \min(mis_A, mis_C, mis_F) = \min(5, 3, 4) = 3$, ta nói: "$Y = \{ACF\}$ là tập mục phổ biến";

Theo ***tính chất 1*** thì các tập con của Y ={ACF} cũng phổ biến, nghĩa là tất cả tập con của Y đều phổ biến – Các con của Y là Y_{sub} = *{(A, 8, 5), (C, 8, 3), (F, 3, 4), (AC, 8, 3), (AF, 3, 4), (CF, 3, 3)}*, tuy nhiên chỉ có các tập mục {A, C, AC, CF} là phổ biến; còn các tập mục {F, AF}, ta có: sup(F) = 3 < mis_F = 4, sup(AF) = 3 < $min(mis_A, mis_F)$ = (5, 4) = 4 là *không phổ biến*. Điều này cho chúng ta thấy: "*Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu thì* ***tính chất 1*** *là không thỏa*".

Tương tự, với Z = {A, F} thì sup(AF) = 3 < $min(mis_A, mis_F)$ = (5, 4) = 4, ta nói: "Tập mục Z = {A, F} *không là tập mục phổ biến*";

Theo *tính chất 2* thì các tập cha của Z ={A, F} cũng *không phổ biến*. Tuy nhiên, ta có Z ={A, F} ⊂ Y = {A, C, F}, mà sup(ACF) = 3 ≥ mis_Y = $min(mis_A, mis_C, mis_F)$ = 3, Y = {A, C, F} là tập mục *phổ biến*. Điều này cho ta thấy "*Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu thì* ***tính chất 2*** *là không thỏa*".

2.3 Tổ chức lưu trữ dữ liệu giao dịch

Lưu trữ dữ liệu giao dịch dạng *bit* là cấu trúc dữ liệu hiệu quả trong khai thác tập phổ biến (Song and Yang, 2008). Chuyển dữ liệu giao dịch thành ma trận nhị phân **BiM**, trong đó mỗi dòng tương ứng với một giao dịch và mỗi cột tương ứng với một mục hàng. Nếu mục hàng thứ i_k xuất hiện trong giao dịch t_j thì *bit* thứ i của dòng t_j mang giá trị *1*, ngược lại sẽ mang giá trị *0*.

Mã giao dịch	A	B	C	D	E	F	G	H
t1	1	0	1	0	1	1	0	0
t2	1	0	1	0	0	0	1	0
t3	0	0	0	0	1	0	0	1
t4	1	0	1	1	0	1	1	0
t5	1	0	1	0	1	0	1	0
t6	0	0	0	0	1	0	0	0
t7	1	1	1	0	1	0	0	0
t8	1	0	1	1	0	0	0	0
t9	1	1	1	0	1	0	1	0
t10	1	0	1	0	1	1	1	0

Hình 1: Dạng bit của dữ liệu giao dịch Ɗ

3 CÁC THUẬT TOÁN

3.1 Tập chiếu và itemset đồng xuất hiện

Tập chiếu của mục hàng i_k trên dữ liệu giao dịch Ɗ: $\pi(i_k)$={t∈ Ɗ | i_k ∈t} là tập các giao dịch có chứa mục hàng i_k (π-*đơn điệu giảm*).

$$sup(i_k) = |\pi(i_k)| \quad (1)$$

Tập chiếu $X = \{i_1, i_2, \dots, i_k\}, \forall i_j \in I (1 \le j \le k)$, $\pi(X) = \pi(i_1) \cap \pi(i_2) \dots \cap \pi(i_k)$.

$sup(X) = |\pi(X)|$ (2)

Ví dụ 3: Bảng 1, có π(A) = {1, 2, 4, 5, 7, 8, 9, 10} và π(B) = {7, 9}. Khi đó, π(AB) = π(A)∩π(B)= {1, 2, 4, 5, **7**, 8, **9**, 10}∩{**7**, **9**} = {**7**, **9**}, π(B) ⊆ π(A) và π(AB) ⊆ π(A).

Định nghĩa 5: Cho $i_k \in I$, ta gọi i_k là *item* hạt nhân. Tập $X_{cooc} \subseteq I$ gọi đồng xuất hiện với i_k: X_{cooc} là tập các *item* xuất hiện cùng i_k thì $\pi(i_k) \equiv \pi(i_k \cup X_{cooc})$. Ký hiệu, $cooc(i_k) = X_{cooc}$.

Ví dụ 4: Xem *item* B là *item* hạt nhân, ta xác định được *itemset* đồng xuất hiện cùng độ phổ biến với *item* B là $cooc$(B) = {A, C, E} và sup(**B**) = sup(**B**ACE) = 2.

Định nghĩa 6: Cho $i_k \in I$, ta gọi i_k là *item* hạt nhân. Tập $Y_{looc} \subseteq I$ chứa các *item* xuất hiện cùng với i_k ít nhất trong một giao dịch, nhưng không đồng xuất hiện: $1 \le |\pi(i_k \cup i_{looc})| < |\pi(i_k)|$, $\forall i_{looc} \in Y_{looc}$. Ký hiệu, $looc(i_k) = Y_{looc}$.

Ví dụ 5: Xem *item* G là *item* hạt nhân, ta xác định được các *item* xuất hiện cùng với *item* B ít nhất trong một giao dịch là $looc$(G) = {B, D, E, F} có π(G) = {2, 4, 5, 9, 10} và π(**G**B) = {9}, π(**G**E) = {5, 9, 10}.

3.2 Thuật toán sinh itemset đồng xuất hiện

Dưới đây là thuật toán sinh các *item* đồng xuất hiện với từng *item* trong dữ liệu giao dịch và lưu trữ vào mảng **Index_COOC**. Mỗi phần tử trong **Index_COOC** gồm 4 thành phần sau:

Index_COOC[j].item: item hạt nhân thứ j;

Index_COOC[j].sup: độ phổ biến của item hạt nhân thứ j;

Index_COOC[j].*cooc*: các *item* đồng xuất hiện cùng *item hạt nhân thứ j* dạng bit;

Index_COOC[j].looc: các item xuất hiện cùng item hạt nhân thứ j ít nhất trong một giao dịch dạng bit;

Mã giả thuật toán 1.

Xây dựng Index_COOC

Đầu vào*:* Dữ liệu giao dịch *Ɗ*

Đầu ra*:* Mảng Index_COOC, ma trận BiM

1. Với mỗi phần tử *j* của mảng **Index_COOC**:
2. **Index_COOC**[j].*item* = *ij*
3. **Index_COOC**[j].*sup* = *0*
4. **Index_COOC**[j].*cooc*= 2^m - 1
5. **Index_COOC**[j].*looc*= 0
6. Với mỗi giao dịch t_i thực hiện:
7. Lưu giao dịch t_i vào ma trận **BiM**

8. Với mỗi item *j* có trong giao dịch t_i thực hiện:
9. ***Index_COOC***[j].*cooc* &= *vectorbit*(t_i)
10. ***Index_COOC***[j].*looc* |= *vectorbit*(t_i)
11. ***Index_COOC***[j].*sup* + +
12. Sắp xếp mảng **Index_COOC** tăng dần theo *sup*
13. Trả về mảng **Index_COOC,** ma trận **BiM**

Từ *dòng 1* đến *dòng 5* là các bước khởi tạo cho mảng **Index_COOC**. *Dòng 6* duyệt dữ liệu giao dịch, ứng với từng giao dịch ta xem xét có chứa *item* thứ *j* thì thực hiện phép toán **AND** trên *bit* để xác định các *item* đồng xuất hiện với *item j* (*dòng 9*) và thực hiện phép toán **OR** trên *bit* để xác định các *item* xuất hiện với *item j* ít nhất trong một giao dịch, nhưng không là đồng xuất hiện (*dòng 10*).

Khởi tạo mảng **Index_COOC**: (thành phần cooc, looc biểu diễn dạng bit) số item là m = 8

item	A	B	C	D	E	F	G	H
sup	0	0	0	0	0	0	0	0
cooc	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
looc	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Duyệt lần lượt từng giao dịch từ t_1 đến t_{10}:

Đọc t_1: {A, C, E, F} có dạng bit là **10101100**

item	A	B	C	D	E	F	G	H
sup	1	0	1	0	1	1	0	0
cooc	**10101100**	11111111	**10101100**	11111111	**10101100**	**10101100**	11111111	11111111
looc	**10101100**	00000000	**10101100**	00000000	**10101100**	**10101100**	00000000	00000000

Duyệt đến t_{10}: {A, C, E, F, G} là **10101110**

item	A	B	C	D	E	F	G	H
sup	8	2	8	2	7	3	5	1
cooc	**10100000**	11101000	**10100000**	10110000	**00001000**	**10100100**	**10100010**	00001001
looc	11111110	11101010	11111110	10110110	11101111	10111110	11111110	00001001

Thuật toán 1, trả về mảng **Index_COOC** sắp tăng theo độ phổ biến của *item* theo Bảng 3.

Bảng 4: Index_COOC sắp tăng thep sup

item	H	B	D	F	G	E	A	C
sup	1	2	2	3	5	7	8	8
cooc	E	A, C, E	A, C	A, C	A, C	Ø	C	A
looc	Ø	G	F, G	D, E, G	B, D, E,F	A, B, C, F, G, H	B, D, E, F, G	B, D, E, F, G

Định nghĩa 7: Cho $i_k \in I$ ($i_1 \prec i_2 \prec \dots \prec i_m$) thứ tự theo độ phổ biến, ta gọi i_k là item hạt nhân. Tập $X_{lexcooc} \subseteq I$ gọi đồng xuất hiện có thứ tự với item i_k: $X_{lexcooc}$ là tập các item xuất hiện cùng i_k và $\pi(i_k) \equiv \pi(i_k \cup X_{lexcooc})$, $i_k \prec i_j$, $\forall i_j \in X_{lexcooc}$. Ký hiệu, $lexcooc(i_k) = X_{lexcooc}$.

Định nghĩa 8: Cho $i_k \in I$ ($i_1 \prec i_2 \prec \dots \prec i_m$) thứ tự theo độ phổ biến, ta gọi i_k là item hạt nhân. Tập $Y_{lexlooc} \subseteq I$ chứa các item xuất hiện có thứ tự cùng với i_k ít nhất trong một giao dịch, nhưng không đồng xuất hiện. Ký hiệu, $lexlooc(i_k) = Y_{lexlooc}$.

$$1 \le \left|\pi(i_k \cup i_{lexlooc})\right| < \left|\pi(i_k)\right|, \ \forall i_{lexlooc} \in Y_{lexlooc}$$

Bổ đề 1: $\forall\ i_k \prec i_j$, nếu $i_j \in lexlooc(i_k)$ thì $sup(i_k \cup i_j) < sup(i_k)$.

Chứng minh: $sup(i_k \cup i_j) < sup(i_k)$, hiển nhiên $\pi(i_k \cup i_j) = \pi(i_k) \cap \pi(i_j) \subset \pi(i_k)$ ■.

Ví dụ 6: Xét *item* B $\prec$ G, G$\in$ *lexlooc*(B)={G}. Ta có, *sup*(**B**G) = 1 < *sup*(**B**)=2.

Bổ đề 2: $lexcooc(i_k) = X_{lexcooc}$ thì $sup(i_k \cup Z_{sub}) = sup(i_k)$, $\forall\ Z_{sub} \subseteq X_{lexcooc}$.

Chứng minh: $lexcooc(i_k) = X_{lexcooc}$, giả sử $X_{lexcooc}$ gồm ℓ *item* thì có $2^{\ell} - 1$ tập con. Với $Z_{sub} \subseteq X_{lexcooc}$ thì ta có $\pi(i_k \cup Y_{sub}) = \pi(i_k) \cap \pi(Y_{sub}) = \pi(i_k)$ ■.

Ví dụ 7: Xét *item* G, với *sup*(G)=5. Ta có, *lexcooc*(**G**) = {A, C} thì 3 *itemset* kết hợp {A, C, AC} và *sup*(**G**) = *sup*(**G**A) = *sup*(**G**C) = *sup*(**G**AC) = 5.

Bổ đề 3: $\forall\ i_k \prec i_j$ và $i_j \in Y_{lexlooc}$: $sup(i_k \cup i_j) = sup(i_k \cup Z_{sub} \cup i_j)$ và $sup(i_k \cup Z_{sub} \cup i_j) < sup(i_k \cup Z_{sub})$, $\forall\ Z_{sub} \subseteq X_{lexcooc}$.

Chứng minh: (theo Bổ đề 2) $\pi(i_k \cup Z_{sub}) = \pi(i_k)$ thì $\pi(i_k \cup Z_{sub} \cup i_j) = \pi(i_k \cup i_j) \subset \pi(i_k)$ ■.

Ví dụ 8: Xét *item* G, với *sup*(G)=5 và i_j = E . Ta có, *lexcooc*(**G**) = {A, C} thì 3 *itemset* kết hợp {A, C, AC} và *sup*(**GE**) = *sup*(**GE**A) = *sup*(**GE**C) = *sup*(**GE**AC) = 3 < *sup*(**G**AC) = 5.

Bổ sung dòng 14, 15 và 16 vào thuật toán 1:

14. Với mỗi phần tử *j* của mảng **Index_COOC**:
15. **Index_COOC**[j].*cooc* = *lexcooc*(i_j)
16. **Index_COOC**[j].*looc*= *lexlooc*(i_j)

Ta có looc(G) = {B, D, E, F} và B, D $\prec$ F $\prec$ G $\prec$ E, nên lexlooc(G) = {E}.

Thực hiện *dòng 14, 15 và 16,* kết quả:

Bảng 5 : Index_COOC chứa itemset đồng xuất hiện có thứ tự

item	H	B	D	F	G	E	A	C
sup	1	2	2	3	5	7	8	8
cooc	E	A, C, E	A, C	A, C	A, C	Ø	C	A
looc	Ø	G	F, G	G, E	E	A, C	Ø	Ø

3.3 Thuật toán tuần tự SEQ-MMSFI

Thuật toán 2 - Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu dựa trên mảng **Index_COOC** chứa các *item* đồng xuất hiện và *xuất hiện ít nhất* trong một giao dịch với *item* hạt nhân (*có thứ tự theo độ phổ biến*).

Định nghĩa 9: Cho $i_k \in I$ ($i_1 \prec i_2 \prec \dots \prec i_m$) thứ tự theo độ phổ biến, $lexcooc(i_k) = X_{lexcooc}$ gồm ℓ *item* thì có $2^\ell - 1$ tập con. Tập chứa các kết hợp $F_{po_{i_k}} = \{i_k \cup Z_j\}, \sup(i_k) = \sup(i_k \cup Z_j)\ \forall Z_j \subset X_{lexcooc}(1 \le j \le 2^\ell - 1)$, gọi là tập chứa các *itemset tiềm năng* theo *item* hạt nhân i_k. Ký hiệu, $potent(i_k) = F_{po_{i_k}}$.

Ngưỡng phổ biến nhỏ nhất của m item $\min\_mis(I) = \min(mis_{i_1}, mis_{i_2}, \dots, mis_{i_m}), \forall i_{j=\overline{1,m}} \in I$.

Tính chất 3: $\forall i_j \in I$ ($1 \le j \le m$), $sup(i_j) < min\_mis(I)$, $sup(i_j) \ge mis_{i_j}$ thì $i_j \in FI$;

Tính chất 4: $\forall i_j \in I$ ($1 \le j \le m$), $sup(i_j) < min\_mis(I)$ thì $i_j \notin FI$;

Bổ đề 4: $\sup(i_k) <$ **min_mis**$(X_{lexcooc}(i_k) \cup i_k)$ và $\sup(i_k) <$ **min_mis**$(Y_{lexlooc}(i_k))$ thì không sinh tập mục phổ biến từ i_k

Chứng minh: (theo Bổ đề 2) $\forall\ Z_{sub} \subseteq X_{lexcooc}$, $sup(i_k \cup Z_{sub}) = sup(i_k) <$ ***min_mis***$(lexcooc(i_k) \cup i_k)$, $2^\ell - 1$ tập con sinh từ *item* hạt nhân i_k là *không phổ biến*; (theo Bổ đề 1) $\forall\ i_k \prec i_j$, nếu $i_j \in lexlooc(i_k)$ thì $sup(i_k \cup i_j) < sup(i_k) <$ ***min_mis***$(Y_{lexlooc}(i_k))$ – các tập mục kết hợp sau i_k cũng *không phổ biến*■.

Ví dụ 9: Xét *item* D, có $lexcooc(D) = \{A, C\} = sup(D) = 2 < min\_mis(lexcooc(D) \cup D) = min(mis_A=5, mis_C=3, mis_D=3) = 3$. Khi đó, các tập mục tiềm năng sinh từ *item hạt nhân* D không phổ biến **F**$_{po}$ ={(**D**, 2, 3), (**DA**, 2, 3), (**DC**, 2, 3), (**DAC**, 2, 3)}. Và $lexlooc(D) = \{F, G\}$ với $sup(D) = 2 < min\_mis(mis_F=4, mis_G=3) = 3$, nên các kết hợp từ *item hạt nhân D* và các tập con của *lexlooc*(D) là **L**$_{sub}$ = {F, G, FG} lần lượt là (**D**F, 1, 3), (**D**G, 1, 3), (**D**FG, 1, 3) không là tập mục phổ biến.

Bổ đề 5: $\sup(i_k) =$ **min_mis**$(X_{lexcooc}, i_k)$ và $\sup(i_k) \le$ **min_mis**$(Z_{sub}), \forall\ Z_{sub} \subset Y_{lexlooc}$ thì $\forall f_j \in$ **F**$_{po}$, $Z_{sub} \cup f_j \notin FI$.

Chứng minh: (theo Bổ đề 1, 2) $\forall\ Z_{sub} \subset Y_{lexlooc}$ thì $\sup(i_k \cup Z_{sub}) = \sup(f_j \cup Z_{sub}) < \sup(i_k) =$ **min_mis**(f_j, Z_{sub})■.

Ví dụ 10: Xét *item* F, có $lexcooc(F) = \{A, C\} = sup(F) = 3 = min\_mis(lexcooc(F), F) = min(mis_A=5, mis_C=3, mis_F=4) = 3$. Khi đó, các tập mục tiềm năng sinh từ *item hạt nhân* F là **F**$_{po}$ ={(**F**A, 3, 4), (**F**C, 3, 3), (**F**AC, 3, 3)}. Và $lexlooc(F) = \{G, E\}$ với **L**$_{sub}$ = {G, E, GE} loại bỏ G vì $mis_G = 3 \ge sup(F)$: $sup(FG)=2 < mis_{FG} = \min(mis_F=4, mis_G=3)=3$.

Mã giả thuật toán 2. Khai thác tập phổ biến SEQ-MMSFI

Đầu vào: Mảng **Dataset**, **Index_COOC** và tập $MIS = \{mis_{i_1}, mis_{i_2}, \dots, mis_{i_m}\}$

Đầu ra**:** Tập phổ biến MMSFI

1. *Loại các item theo tính chất 4 và Bổ đề 4*
2. Với mỗi Index_COOC[k].*sup* ≥ *min_mis* = $min(mis_{i_1}, mis_{i_2}, \dots, mis_{i_m})$
3. Nếu Index_COOC[k].sup ≥ mis_{i_k}
4. FI[k] = FI[k] ∪{i_k}//(*tính chất 3*)
5. *Co* = Index_COOC[k].*cooc*
6. *Lo* = Index_COOC[k].*looc*
7. $C_{sub} \leftarrow$ *các tập con của Co*
8. Với mỗi *itemset* $IS_i \in C_{sub}$
9. $F_{po}[k] = F_{po}[k] \cup \{i_k \cup IS_i\}$
10. $L_{sub} \leftarrow$ *các tập con của Lo*
11. Nếu (Index_COOC[k].*sup*=*min_mis*(*Co*)) thì
12. $L_{sub} \leftarrow L_{sub} \setminus \{\forall z_{sub} \in L_{sub} \mid$ Index_COOC[k].*sup*≤*min_mis*(z_{sub})$\}$
13. $F_{sub} \leftarrow$ *các kết hợp giữa* L_{sub} *và* i_k
14. Với mỗi $f_i \in F_{po}$
15. Với mỗi $f_j \in F_{sub}$
16. FI[k] = FI[k]∪{$f_i \cup f_j$}
17. FI[k] = FI[k]∪$F_{po}[k]$
18. Sắp xếp FI giảm dần theo *sup*
19. Trả về tập phổ biến MMSFI

Ví dụ 11: Cho dữ liệu giao dịch Ɗ trong Bảng 1 và tập các ngưỡng phổ biến tối thiểu của từng *item* theo Bảng 2. Sau khi thực hiện thuật toán 1, ta có mảng chứa các *itemset* đồng xuất hiện như Bảng 5.

Ta có: $MIS = \{mis_A=5, mis_B=2, mis_C=3, mis_D=3, mis_E=2, mis_F=4, mis_G=3, mis_H=2\}$ và ***min_mis(I)*** = 2;

Dòng 1, loại các *item* theo tính chất 4 – có *item* H; theo Bổ đề 4 – có *item* D ;

Với *item* H, có $sup(H) = 1 <$ ***min_mis*** : loại bỏ *item* H khỏi danh sách các *item* khai phá; (*tính chất 4*)

Xét *item* D là *item* cơ sở – $sup(D) = 2 <$ ***min_mis***(lexcooc(D), lexlooc(D))= ***min_mis***($mis_A=5, mis_C=3, mis_F=4, mis_G=3$)=$mis_D$ =

3. $\mathbf{FI}_{[D]} = \{\varnothing\}$: loại bỏ *item* D khỏi danh sách các *item* khai phá; (*bổ đề 4*)

Xét *item* B có *lexcooc*(B) ={E, A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ = {(**B**, 2, 2), (**BE**, 2, 2), (**BA**, 2, 2), (**BC**, 2, 2), (**BEA**, 2, 2), (**BEC**, 2, 2), (**BAC**, 2, 2), (**BEAC**, 2, 2)}. Với **Lo** = {G} và $\mathbf{L_{sub}} = \{\varnothing\}$ (*dòng 11, 12*), $\mathbf{F_{sub}} = \{\varnothing\}$. Ta có, $\mathbf{FI_{[B]}}$ = {(**B**, 2, 2), (**BE**, 2, 2), (**BA**, 2, 2), (**BC**, 2, 2), (**BEA**, 2, 2), (**BEC**, 2, 2), (**BAC**, 2, 2), (**BEAC**, 2, 2)}.

Xét *item* F có *lexcooc*(F) = {A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**FA**, 3, 4), (**FC**, 3, 3), (**FAC**, 3, 3)}. Với **Lo**={G, E}, $\mathbf{L_{sub}}$ = {~~G~~, E, GE} (*dòng 11, 12*), và sinh $\mathbf{F_{sub}}$ ={(**FE**, 2, 2), (**FGE**, 1, 2)}. Sinh tập $\mathbf{FI}_{[F]}$ ={(**FC**, 3, 3), (**FE**, 2, 2), (**FAC**, 3, 3), (**FEA**, 2, 2), (**FEC**, 2, 2), (**FEAC**, 2, 2)}.

Xét *item* G có *lexcooc*(G) = {A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**G**, 5, 3), (**GA**, 5, 3), (**GC**, 5, 3), (**GAC**, 5, 3)}. Với **Lo** = {E}, $\mathbf{L_{sub}}$ ={E} và sinh $\mathbf{F_{sub}}$ = {(**GE**, 3, 2)}. Sinh tập phổ biến $\mathbf{FI}_{[G]}$ ={(**G**, 5, 3), (**GA**, 5, 3), (**GC**, 5, 3), (**GAC**, 5, 3), (**GE**, 3, 2), (**GEA**, 3, 2), (**GEC**, 3, 2), (**GEAC**, 3, 2)}.

Xét *item* E có *lexcooc*(E) ={$\varnothing$}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**E**, 7, 2)}. Với **Lo** ={A, C}, $\mathbf{L_{sub}}$ ={A, C, AC} và sinh $\mathbf{F_{sub}}$ ={(**EA**, 5,2), (**EC**, 5, 2), (**EAC**, 5, 2)}. Sinh tập phổ biến $\mathbf{FI}_{[E]}$ ={(**E**, 7, 2), (**EA**, 5,2), (**EC**, 5, 2), (**EAC**, 5, 2)}.

Xét *item* A, có *lexcooc*(A) = {C}: sinh tập tiềm năng $\mathbf{F_{po}}$ = {(**A**, 8, 5), (**AC**, 8, 3)}, **Lo** = {$\varnothing$}, $\mathbf{L_{sub}}$ = {$\varnothing$}, $\mathbf{F_{sub}}$ = {$\varnothing$}. Sinh tập phổ biến khi xét *item* A là $\mathbf{FI}_{[A]}$ = {(**A**, 8, 5), (**AC**, 8, 3)}.

Xét *item* C, có *lexcooc*(C) = {$\varnothing$}, $\mathbf{F_{po}}$ = {(**C**, 8, 3)}, **Lo** = {$\varnothing$}, $\mathbf{L_{sub}}$ = {$\varnothing$}, $\mathbf{F_{sub}}$ = {$\varnothing$}. Sinh tập phổ biến khi xét *item* C: $\mathbf{FI}_{[C]}$ = {(**C**, 8, 3)}.

Tập phổ biến **MMSFI** trên dữ liệu giao dịch $\mathcal{D}$ ở Bảng 1 và ngưỡng phổ biến tối thiểu của từng item trong Bảng 2.

Bảng 6: Tập phổ biến trên $\mathcal{D}$

$\mathbf{FI}_{[B]}$	(**B**,2,2)	(**BE**,2,2)	(**BA**,2,2)	(**BC**,2,2)	(**BEA**,2,2)	(**BEC**,2,2)	(**BAC**,2,2)	(**BEAC**,2,2)
$\mathbf{FI}_{[F]}$	(**FC**,3,3)	(**FE**,2,2)	(**FAC**,3,3)	(**FEA**,2,2)	(**FEC**,2,2)	(**FEAC**,2,2)		
$\mathbf{FI}_{[G]}$	(**G**,5,3)	(**GA**,5,3)	(**GC**,5,3)	(**GE**,3,2)	(**GAC**,5,3)	(**GEA**,3,2)	(**GEC**,3,2)	(**GEAC**,3,2)
$\mathbf{FI}_{[E]}$	(**E**,7,2)	(**EA**,5,2)	(**EC**,5,2)	(**EAC**,5,2)				
$\mathbf{FI}_{[A]}$	(**A**,8,5)	(**AC**,8,3)						
$\mathbf{FI}_{[C]}$	(**C**,8,3)							

4 THUẬT TOÁN MCP-MMSFI

Ngày nay, nhiều máy tính cá nhân và máy trạm có trên hai nhân – BXLĐN, cho phép nhiều luồng xử lý (*thread*) được thực hiện đồng thời – điều này làm cho các máy tính có được tốc độ xử lý nhanh hơn và khả năng đa nhiệm tốt hơn. Để tận dụng hết khả năng của BXLĐN cần phân phối xử lý đồng thời trên nhiều nhân cho nhiều pha/bài toán khác nhau để tiết kiệm thời gian và nâng cao hiệu suất.

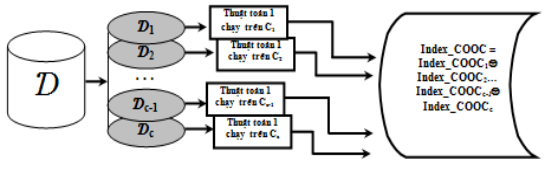
Chúng tôi xây dựng thuật toán song song **MCP-MMSFI** khai thác tập phổ biến trên BXLĐN dựa trên **SEQ-MMSFI**.

Thuật toán tuần tự **SEQ-MMSFI**, có *2 pha*:

– **Pha 1**: Xây dựng mảng **Index_COOC**;

– **Pha 2**: Thuật toán **SEQ-MMSFI** khai thác tập phổ biến từ mảng **Index_COOC**.

Bước thứ nhất, song song hóa **Pha 1**:



Hình 2: Sơ đồ song song hóa cho Pha 1

Hình 2, phân chia dữ liệu $\mathcal{D}$ thành c dữ liệu con $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{c-1}, \mathcal{D}_c$ ứng với từng nhân C_i thực hiện thuật toán 1 với đầu vào là dữ liệu $\mathcal{D}_i$ và đầu ra là mảng $Index\_COOC_i$ tương ứng. Để tính mảng **Index_COOC** cho $\mathcal{D}$:

Index_COOC=Index_COOC$_1$⊕Index_COOC$_2$⊕…

Sau đó, sắp mảng **Index_COOC** tăng dần theo *sup* và chuẩn hóa theo dòng 14, 15 và 16.

Ví dụ 12: Giả sử $\mathcal{D}$ được chia thành 2 tập - $\mathcal{D}_1$ có 5 giao dịch {t1, t2, t3, t4, t5} và $\mathcal{D}_2$ có 5 giao dịch {t6, t7, t8, t9, t10}.

$\mathcal{D}_1$ chạy trên nhân C_1 trả về $Index\_COOC_1$:

item	A	B	C	D	E	F	G	H
sup	4	0	4	1	3	2	3	1
cooc	10100000	11111111	10100000	10110110	00001000	10100100	10100010	00001001
looc	10111110	*00000000*	10111110	10110110	10101111	10111110	10111110	00001001

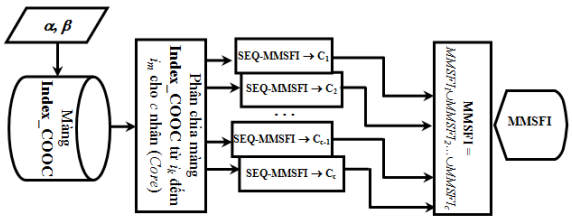
$\mathcal{D}_2$ chạy trên nhân C_2 trả về $Index\_COOC_2$:

item	A	B	C	D	E	F	G	H
sup	4	2	4	1	4	1	2	0
cooc	10100000	11101000	10100000	10110000	00001000	10101110	10101010	11111111
looc	11111110	11101010	11111110	10110000	11101110	10101110	11101110	00000000

Tính mảng **Index_COOC** cho $\mathcal{D}$, :**Index_COOC**=$Index\_COOC_1$⊕$Index\_COOC_2$

item	A	B	C	D	E	F	G	H
sup	8	2	8	2	7	3	5	1
cooc	10100000	11101000	10100000	10110000	00001000	10100100	10100010	00001001
looc	11111110	11101010	11111110	10110110	11101111	10111110	11111110	00001001

Bước thứ hai, song song hóa **Pha 2**:



Hình 3 : Sơ đồ song song hóa cho *Pha 2*

Hình 3 phân chia mảng **Index_COOC** từ i_k đến i_m thành c phần ứng với từng nhân C_j thực hiện thuật toán **SEQ-MMSFI** với đầu vào là mảng **Index_COOC** từ phần tử thứ $k+(j-1)*((m-k+1)\ div\ c)$ đến phần tử thứ $k+j*((m-k+1)\ div\ c)$ và đầu ra là tập phổ biến $\boldsymbol{MMSFI_j}$ tương ứng. Tập phổ biến cho $\mathcal{D}$:

$\mathbf{MMSFI_D} = MMSFI_1 \cup MMSFI_2 \ldots \cup MMSFI_c$

Ví dụ 13: Cho $\mathcal{D}$ trong Bảng 1, Bảng 2. Sau khi thực hiện song song hóa Pha 1, ta có mảng chứa các *itemset* đồng xuất hiện như Bảng 4.

Nhân C_1 chạy thuật toán **SEQ-MMSFI** từ *item* B đến G và sinh tập phổ biến $MMSFI_1$:

$FI_{[B]}$	(B,2,2)	(BE,2,2)	(BA,2,2)	(BC,2,2)	(BEA,2,2)	(BEC,2,2)	(BAC,2,2)	(BEAC,2,2)
$FI_{[F]}$	(FC,3,3)	(FE,2,2)	(FAC,3,3)	(FEA,2,2)	(FEC,2,2)	(FEAC,2,2)		
$FI_{[G]}$	(G,5,3)	(GA,5,3)	(GC,5,3)	(GE,3,2)	(GAC,5,3)	(GEA,3,2)	(GEC,3,2)	(GEAC,3,2)

Nhân C_2 chạy thuật toán SEQ-**MMSFI** từ *item* E đến C và sinh tập phổ biến $MMSFI_2$:

$FI_{[E]}$	(E,7,2)	(EA,5,2)	(EC,5,2)	(EAC,5,2)
$FI_{[A]}$	(A,8,5)	(AC,8,3)		
$FI_{[C]}$	(C,8,3)			

Tập phổ biến **MMSFI** trên dữ liệu giao dịch $\mathcal{D}$ và tập ngưỡng ở Bảng 2, được tính $\mathbf{MMSFI_D} = MMSFI_1 \cup MMSFI_2$ như Bảng 6.

5 KẾT QUẢ THỰC NGHIỆM

Thực nghiệm trên CF-74 (2 *core*, 2 *thread*), Core Duo 2.0 GHz, 4GB RAM, MSVC#2010.

Thực nghiệm trên hai nhóm dữ liệu:

Nhóm dữ liệu thực có mật độ dày: từ kho dữ liệu về học máy của trường Đại học California (Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science) gồm 2 tập **Chess** và **Mushroom**.

Nhóm dữ liệu giả lập có mật độ thưa: sử dụng phần mềm phát sinh dữ liệu giả lập của trung tâm nghiên cứu IBM Almaden (IBM Almaden Research Center, San Joe, California 95120, U.S.A [http://www.almaden.ibm.com]) gồm 2 tập **T10I4D100K** và **T40I10D100K**.

Bảng 7: Dữ liệu thực nghiệm

$\mathcal{D}$	Số item	Số item nhỏ nhất/giao dịch	Số item lớn nhất/ giao dịch	Số item trung bình/ giao dịch	Mật độ (%)
Chess	75	37	37	37	49,3%
Mushroom	119	23	23	23	19,3%
T10I4D100K	870	1	29	10	1,1%
T40I10D100K	942	4	77	40	4,2%

Kết quả thực nghiệm: để nhất quán khi so sánh với các thuật toán trước, nhóm tác giả sử dụng phương thức gán các giá trị ngưỡng phổ biến tối thiểu cho từng mục hàng theo các thuật toán (Liu *et al.*, 1999; Hu *et al.*, 2006; Kiran *et al.*, 2011). Gán các giá trị ngưỡng phổ biến tối thiểu cho từng mục hàng theo công thức:

$$mis_{i_j} = \mathrm{MAX}\left(\alpha, \beta \times \sup(i_j)\right), \forall i_{j=\overline{1,m}} \in I \quad (3)$$

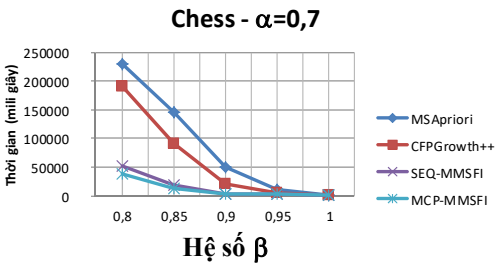
Trong đó, giá trị α là ngưỡng phổ biến nhỏ nhất có thể. Hệ số $\beta (0 \le \beta \le 1)$ là hệ số điều khiển, nhằm xác định giá trị các ngưỡng phổ biến tối thiểu của từng mục hàng theo độ phổ biến của từng mục hàng. Trường hợp $\beta=0$, lúc này trở thành bài toán khai thác tập phổ biến với *một* ngưỡng phổ biến tối thiểu là α. Dưới đây, chúng tôi so sánh thuật toán đề xuất **SEQ-MMSFI**, thuật toán song song **MCP-MMSFI** với thuật toán **MSApriori**, **CFPGrowth++**.

Hiệu suất thực hiện thuật toán song song **MCP-MMSFI** trên BXLĐN:

$$HS = 1 - \left(T_M - \frac{T_S}{c}\right) / \left(\frac{T_S}{c}\right) \quad (4)$$

Trong đó:

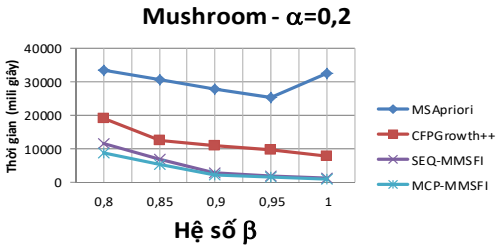
- T_S: thời gian thực hiện tuần tự
- T_M: thời gian thực hiện song song
- c: số lượng nhân của CPU (số *core*)



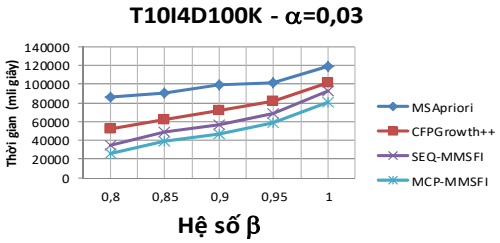
Hình 4: Thời gian khai thác MMSFI trên Chess

Hình 4 và 5 là kết quả thực nghiệm trên tập dữ liệu **Chess** và **Mushroom** có mật độ cao, ta thấy thuật toán tuần tự **SEQ-MMSFI** nhanh hơn

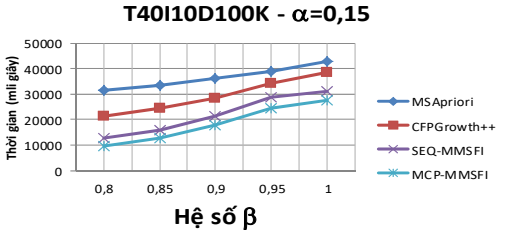
MSApriori, **CFPGrowth++** và thuật toán chạy trên BXLĐN là **MCP-MMSFI** có thời gian thực hiện nhanh hơn thuật toán tuần tự **SEQ-MMSFI.** Hiệu suất trung bình của thuật toán **MCP-MMSFI** lần lượt: (*78%; độ lệch chuẩn 4,8%*) và (*79%; độ lệch chuẩn 3,2%*).



Hình 5: Thời gian khai thác MMSFI trên Mushroom



Hình 6: Thời gian khai thác MMSFI trên T10I4D100K



Hình 7: Thời gian khai thác MMSFI trên T40I10D100K

Hình 6 và 7 là kết quả thực nghiệm trên tập dữ liệu giả lập có mật độ thấp **T10I4D100K** và **T40I10D100K,** ta thấy thuật toán tuần tự **SEQ-MMSFI** nhanh hơn **MSApriori**, **CFPGrowth++** và thuật toán chạy trên BXLĐN là **MCP-MMSFI** có thời gian thực hiện nhanh hơn thuật toán tuần tự **SEQ-MMSFI.** Hiệu suất trung bình của **MCP-MMSFI** lần lượt: (*82%; độ lệch chuẩn 4,4%*) và (*82%; độ lệch chuẩn 5,5%*).

Kết quả trên cho thấy thuật toán khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu **MCP-MMSFI** trên BXLĐN tốt hơn rất nhiều so với thuật toán **MSApriori**, **CFPGrowth++**. Thuật toán **MCP-MMSFI** cần được thực nghiệm thêm trên các dữ liệu cỡ lớn và so sánh thêm với các thuật toán chạy trên hệ thống toán phân tán Hadoop, Spark.

6 KẾT LUẬN

Nhóm tác giả đã đề xuất kiến trúc tuần tự khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu **SEQ-MMSFI**. Với kiến trúc như trên, khi người dùng khai thác tập phổ biến với *bộ ngưỡng khác* thì thuật toán đề xuất chỉ thực hiện khai thác tập phổ biến trên mảng **Index_COOC** đã tính ở lần khai thác trước làm giảm thời gian xử lý đáng kể. Từ thuật toán tuần tự **SEQ-MMSFI**, chúng tôi mở rộng và song song hóa thực hiện trên BXLĐN gọi là thuật toán **MCP-MMSFI**. Hiệu suất trung bình khi song song hóa là 80% và độ lệch chuẩn 4,6% (*trên dữ liệu thực nghiệm*).

Tương lai, nhóm tác giả sẽ mở rộng thuật toán **MCP-MMSFI** để có thể khai thác nhanh tập phổ biến với nhiều ngưỡng phổ biến tối thiểu trên hệ thống điện thoại thông minh đa lõi có tài nguyên hạn chế, cũng như mở rộng trên hệ thống phân tán như Hadoop, Spark.

LỜI CẢM TẠ

Nhóm tác giả xin cảm ơn sự hỗ trợ từ Trường Đại học Khoa học Xã hội và Nhân văn, ĐHQG-HCM.

TÀI LIỆU THAM KHẢO

Agrawal, R., Imilienski, T., Swami, A., 1993. Mining association rules between sets of large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC: 207-216.

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. Proceedings of International Conference on Very Large Data Base, Santiago, Chile: 478-499.

Han, J., Pei, J., Yin, Y., Mao, R., 2004. Mining frequent patterns without candidate generation: A frequent pattern tree approach. Data Mining and Knowledge Discovery, 8(1): 53–87.

Hu, Y.H., Chen, Y.L., 2006. Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism. Decision Support Systems, 42(1): 1–24.

Kiran, R. U., Reddy, P. K., 2011. Novel techniques to reduce search space in multiple minimum supports based frequent pattern mining algorithms. In EDBT: 11–20.

Liu, B., Hsu, W., Ma, Y., 1999. Mining association rules with multiple minimum supports. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge discovery and Data mining:337–341.

Song, W., Yang, B., 2008. Index-BitTableFI: An improved algorithm for mining frequent itemsets. Knowledge-Based Systems 21: 507-513.