TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP

NGHIÊN CỨU VỀ THUẬT GIẢI DI TRUYỀN GA VÀ ỨNG DỤNG

GIẢNG VIÊN HƯỚNG DẪN: ThS. Phạm Đức Thành
SINH VIÊN THỰC HIỆN: Đỗ Anh Duy – 20DH111943

TP. HỒ CHÍ MINH, NGÀY 23 THÁNG 05 NĂM 2024

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP

NGHIÊN CỨU VỀ THUẬT GIẢI DI TRUYỀN GA VÀ ỨNG DỤNG

GIẢNG VIÊN HƯỚNG DẪN: ThS. Phạm Đức Thành

SINH VIÊN THỰC HIỆN: Đỗ Anh Duy – 20DH111943

TP. HỒ CHÍ MINH, NGÀY 23 THÁNG 05 NĂM 2024

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn chân thành đến tất cả những người đã hỗ trợ và đóng góp cho quá trình hoàn thành khóa luận này.

Đầu tiên và quan trọng nhất, em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Phạm Đức Thành - người đã dành thời gian và kiến thức của mình để hỗ trợ và chỉ dẫn em qua mỗi bước trong quá trình nghiên cứu và viết khóa luận.

Em cũng xin gửi lời cảm ơn sâu sắc đến Ban Giám Hiệu nhà trường đã tạo điều kiện cho em được học tập và rèn luyện trong môi trường giáo dục chất lượng cao. Nhà trường và khoa đã trang bị cho em đầy đủ kiến thức và kỹ năng cần thiết để có thể hoàn thành tốt công việc học tập, đặc biệt là trong ngành học công nghệ thông tin.

Em vô cùng vinh dự được trình bày Khóa luận Tốt nghiệp này. Việc hoàn thành được khóa luận này là kết quả của sự nỗ lực không ngừng nghỉ của bản thân em và sự giúp đỡ nhiệt tình từ quý thầy cô, gia đình và bạn bè.

Em ý thức được rằng khóa luận của em còn có nhiều thiếu sót, mong nhận được sự góp ý quý báu từ quý thầy cô và các bạn để em có thể hoàn thiện hơn nữa. Lời cảm ơn của em không đủ để bày tỏ tất cả lòng biết ơn của mình.

Em hy vọng rằng công trình này có thể là một cống hiến nhỏ bé nhưng ý nghĩa đối với cộng đồng nghiên cứu và giáo dục.

Em xin chân thành cảm ơn!

Đỗ Anh Duy

LỜI CAM ĐOAN

Em xin cam đoan khóa luận tốt nghiện nghiên cứu thuật giải di truyền và ứng dụng trong lĩnh vực khai thác tập mục hữu ích cao là do bản thân em dưới sự hướng dẫn của thầy Ths. Phạm Đức Thành.

Toàn bộ nội dung là kết quả nghiên cứu của bản thân em, không sao chép hay vi phạm bản quyền của bất kỳ ai.

Các số liệu và thông tin trong khóa luận đều được thu thập tin cậy. Em đã hoàn thành đầy đủ các yêu cầu về học tập và đạo đức trong quá trình thực hiện khóa luận.

Em xin chịu trách nhiệm trước hội đầu và nhà trường về nội dung những cam đoan trên.

Đỗ Anh Duy

23/05/2024

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH VÀ BIỂU ĐỒ

DANH MỤC CHỮ VIẾT TẮT

STT	Chữ viết tắt	Mô tả
1	NST	Nhiễm sắc thể
2	GA	Genetic algorithms (Thuật giải di truyền)
3	HUI	High utility itemset (Tập mục hữu ích cao)
4	HUIM	High utility itemset mining (Khai thác tập mục hữu ích cao)
5	FHM	Faster High utility itemset mining
6	CSDL	Cơ sở dữ liệu
7	DB	Database (Cơ sở dữ liệu)
8	IU	Internal Utility (Hữu ích nội)
9	EU	External Utility (Hữu ích ngoại)
10	TU	Transaction Utility (Hưu ích của giao tác)
11	TWU	Transaction weight utility (Hữu ích theo trọng số)
12	U	Utility (Hữu ích)
13	TID	Transaction ID (Mã giao tác)
14	RU	Remaning utility (Hữu ích còn lại)
15	&	Phép AND trên bit
16	STT	Số thứ tự
17	MU	Min utility (Hữu ích tối thiểu)
18	CP	Crossover probability (Tỷ lệ lai ghép)
19	MP	Mutation probability (Tỷ lệ đột biến)

19	GEN	Generations (Thế hệ)
20	PS	Population size (Kích thước quần thể)
21	OS	Operating system (Hệ điều hành)
22	RAM	Ramdom access memory

CHƯƠNG 1. MỞ ĐẦU

1.1. Giới thiệu về đề tài

Khai phá dữ liệu là ngành khoa học đang ngày được quan tâm nghiên cứu và phát triển do những ứng dụng thiết thực mà nó mang lại. Khai phá dữ liệu là phần cốt lõi của phát hiện tri thức, trong khai phá dữ liệu phát hiện các luật là một trong những nội dung cơ bản và phổ biến nhất. Các phương pháp phát hiện luật nhằm tìm ra sự phụ thuộc giữa các tính chất của các đối tượng hay các thuộc tính trong cơ sở dữ liệu.

Trong thời đại số hóa ngày nay, dữ liệu là một nguồn tài nguyên vô cùng quý báu. Từ thông tin cá nhân đến dữ liệu thương mại, mọi thứ đều được tạo ra, chia sẻ và lưu trữ trong hệ thống thông tin to lớn. Tuy nhiên, với khối lượng lớn dữ liệu này, việc phân tích và trích xuất thông tin hữu ích trở nên phức tạp. Trong ngữ cảnh này, khai thác tập hữu ích cao (frequent high utility itemset mining) đã trở thành một lĩnh vực nghiên cứu quan trọng trong khoa học máy tính và dữ liệu lớn.

Dựa vào những cơ sở trên, bài luận này sẽ tập trung tìm hiểu một số thuật toán hay hướng tiếp cận khai khai thác tập hữu ích cao thông qua thuật giải di truyền.

Dù đã cố gắng hết sức, nhưng do hạn chế về thời gian và tài liệu, cũng như kiến thức, bài luận của em vẫn còn nhiều thiếu sót. Em rất mong nhận được sự góp ý chân thành từ các thầy cô để có thể hoàn thiện bài làm hơn nữa.

1.2. Mục tiêu và ý nghĩa

Mục tiêu của nghiên cứu này là khám phá và ứng dụng các phương pháp khai thác tập hữu ích cao để tạo ra giá trị trong các lĩnh vực cuộc sống hiện nay. Bằng cách áp dụng các kỹ thuật phân tích dữ liệu tiên tiến, nghiên cứu này nhằm giải quyết các vấn đề thực tế và cung cấp những thông tin hữu ích cho các tổ chức và cá nhân.

1.3. Phương pháp nghiên cứu dự kiến

- Thuật giải di truyền (GA): Phát triển và triển khai thuật toán di truyền để tối ưu hóa các tập hữu ích cao đã khai thác. Quá trình di truyền bao gồm các phép lai

ghép, đột biến và lựa chọn để tạo ra các thế hệ con cái mới có khả năng cao hơn trong việc khai thác tập hữu ích cao.

- Tổng quan về khai phá dữ liệu: Trình bày về khái niệm, mục đích và ứng dụng của việc khai phá dữ liệu.
- Tổng quan về khai phá luật kết hợp: Trình bày các khái niệm cơ bản, các thuật toán về khai thác tập phổ biến và sinh luật kết hợp, và các ứng dụng trong thực tế.
- Tổng quan về tập hữu ích cao: Trình bày các khái niệm cơ bản, các thuật toán phổ biến và các ứng dụng trong thực tế về khai thác tập hữu ích cao.
- Đánh giá hiệu suất: Đánh giá hiệu suất của các phương pháp và công cụ được phát triển thông qua các thử nghiệm và so sánh với các phương pháp hiện có.

1.4. Kết quả dự kiến và ứng dụng thực tế

Kết quả của nghiên cứu này dự kiến sẽ cung cấp một cái nhìn tổng quan về cách kết hợp thuật giải di truyền và các phương pháp tối ưu để có thể tìm ra tập hữu ích cao một cách hiệu quả để tạo ra giá trị trong các lĩnh vực thực tiễn. Nghiên cứu được phát triển có thể được sử dụng để tối ưu hóa quy trình kinh doanh, cải thiện dịch vụ khách hàng và đưa ra quyết định chiến lược trong nhiều lĩnh vực như khuyến mãi và bán chéo, phân tích giỏ hàng, quản lý tồn kho, phát hiện gian lận, phân tích rủi ro, …

1.5. Kế hoạch nghiên cứu

- Tiến hành nghiên cứu các khái niệm các hàm vệ tinh, hàm chính của thuật toán, sơ đồ hoạt động, cách triển khai và phát triển thuật giải di truyền bằng ngôn ngữ lập trình Python.
- Tìm hiểu và nghiên cứu khái niệm về thuật toán Apriori để khai phá luật kết hợp trong cơ sở dữ liệu giao tác. Đánh giá ứng dụng và hạn chế của tập phổ biến trong thực tế ngày nay.
- Tìm hiểu và nghiên cứu các khái niệm, công thức, ưu điểm, hạn chế, … của việc khai thác tập hữu ích cao từ đó đưa ra hướng giải quyết những hạn chế của việc khai thác dữ liệu này.

- Triển khai một số thuật toán khai thác tập hữu ích cao phổ biến như HUI – Miner, FHM, …
- Áp dụng, triển khai và phát triển thuật giải di truyền để phù hợp, tối ưu hóa trong việc khai thác các tập hữu ích cao từ dữ liệu lớn.
- Tích hợp lập trình song song để cải thiện các hàm tính toán, giúp tăng tốc độ trả về kết quả của việc triển khai thuật toán trên tập dữ liệu lớn.
- Đánh giá hiệu suất của hệ thống được phát triển thông qua các thử nghiệm và so sánh với các phương pháp hiện có.

1.6. Kết luận

Kết hợp khai thác tập hữu ích cao và thuật toán di truyền là một cách tiếp cận tiềm năng để tạo ra giá trị từ dữ liệu lớn. Nghiên cứu này hy vọng sẽ đóng góp vào sự phát triển của lĩnh vực khai phá dữ liệu và ứng dụng thực tiễn của tập phổ biến, tập hữu ích cao. Đồng thời cũng mở ra những cơ hội mới trong việc áp dụng dữ liệu trong cuộc sống hàng ngày.

CHƯƠNG 2. TỔNG QUAN

2.1. Khái niệm về thuật toán và thuật giải

2.1.1. Thuật toán

Một thuật toán (hay giải thuật) là một thủ tục để giải quyết một bài toán hay một vấn đề, bằng cách thực thi một dãy hữu hạn thao tác. Thuật toán có thể thực hiện các công việc như tính toán, xử lý dữ liệu, suy luận logic tự động…

Đặc trừng của thuật toán:

- Tính chính xác: Các bước thực thi phải được phát biểu chính xác, chặt chẽ.
- Tính duy nhất: Kết quả của mỗi bước được định nghĩa một cách duy nhất và chỉ phụ thuộc vào nhập liệu và kết quả của các bước trước đó.
- Tính hữu hạn: Thuật toán ngừng sau một khi một số hữu hạn các chỉ thị lệnh được thực hiện.
- Nhập liệu (input): – thuật toán nhận nhập liệu (có thể trống).
- Xuất liệu (output) – thuật toán tạo ra xuất liệu.
- Tính tổng quát – thuật toán có thể được vận dụng cho một tập hợp nhập liệu.

Các phương pháp biểu diễn thuật toán:

- Bằng ngôn ngữ tự nhiên.
- Bằng lưu đồ (flowchart).
- Bằng mã giả (pseudo code).

2.1.2. Thuật giải

Thuật giải là khái niệm thuật toán được mở rộng để có thể đi đến lời giải chấp nhận được nhưng có độ phức tạp tốt hơn (nhiều) so với thuật toán (nếu tồn tại).

Có nhiều bài toán cho đến nay vẫn chưa tìm ra một cách giải theo kiểu thuật toán và cũng không biết là có tồn tại thuật toán hay không. Có nhiều bài toán đã có thuật toán để giải nhưng không chấp nhận được vì thời gian giải theo thuật toán đó quá lớn hoặc các điều kiện cho thuật toán khó đáp ứng. Có những bài toán được giải theo những cách giải vi phạm thuật toán nhưng vẫn chấp nhận được.

Tính đúng của thuật toán bây giờ không còn bắt buộc đối với một số cách giải bài toán, nhất là các cách giải gần đúng. Trong thực tiễn có nhiều trường hợp người ta chấp nhận các cách giải thường cho kết quả tốt nhưng ít phức tạp và hiệu quả. Khái niệm thuật toán được mở rộng như trên gọi là thuật giải. Và Thuật giải thường được dùng nhiều trong trí tuệ nhân tạo.

2.2. Thuật giải di truyền (GA)

2.2.1. Khái niệm về thuật giải di truyền

Thuật giải di truyền (GA) là một kỹ thuật chung giúp giải quyết tìm kiếm giải pháp (lời giải) tương đối tối ưu dựa trên cơ chế chọn lọc và di truyền tự nhiên.

GA mô phỏng lại quá trình tiến hóa của các quần thể sinh học theo thuyết tiến hóa Darwin, sử dụng các nguyên lý như di truyền, bột biến, chọn lọc tự nhiên và lai chéo để tìm giải pháp tối ưu cho các bài toán phức tạp [2].

2.2.2. Cách thức hoạt động

a) Khởi tạo quần thể ban đầu (Initial Population)

Tạo ra một tập hợp quần thể ban đầu gồm các giải pháp tiềm năng (cá thể). Mỗi giải pháp được biểu diễn bằng một chuỗi mã (NST).

b) Đánh giá (Fitness)

Tùy vào hàm mục tiêu của bài toán cụ thể, ta sẽ có hàm đánh giá độ thích nghi của cá thể.

Tính độ thích nghi (xác định giá trị mục tiêu) của mỗi giải pháp trong quần thể.

Độ thích nghi được xác định bởi một hàm mục tiêu (fitness), thể hiện mức độ tốt mà giải pháp đáp ứng được yêu cầu của bài toán.

c) Chọn lọc (Selection)

Lựa chọn các giải pháp tốt nhất trong quần thể đã được đánh giá trước đó để tạo ra thế hệ tiếp theo. Những giải pháp có độ thích nghi cao hơn có nhiều khả năng được chọn cao hơn.

d) Lai ghép (Crossover)

Kết hợp các nhiễm sắc thể (giải pháp) được chọn lọc ngẫu nhiên để tạo ra những NST mới cho thế hệ tiếp theo. Quá trình này mô phỏng lại quá tình sinh sản hữu tính trong tự nhiên.

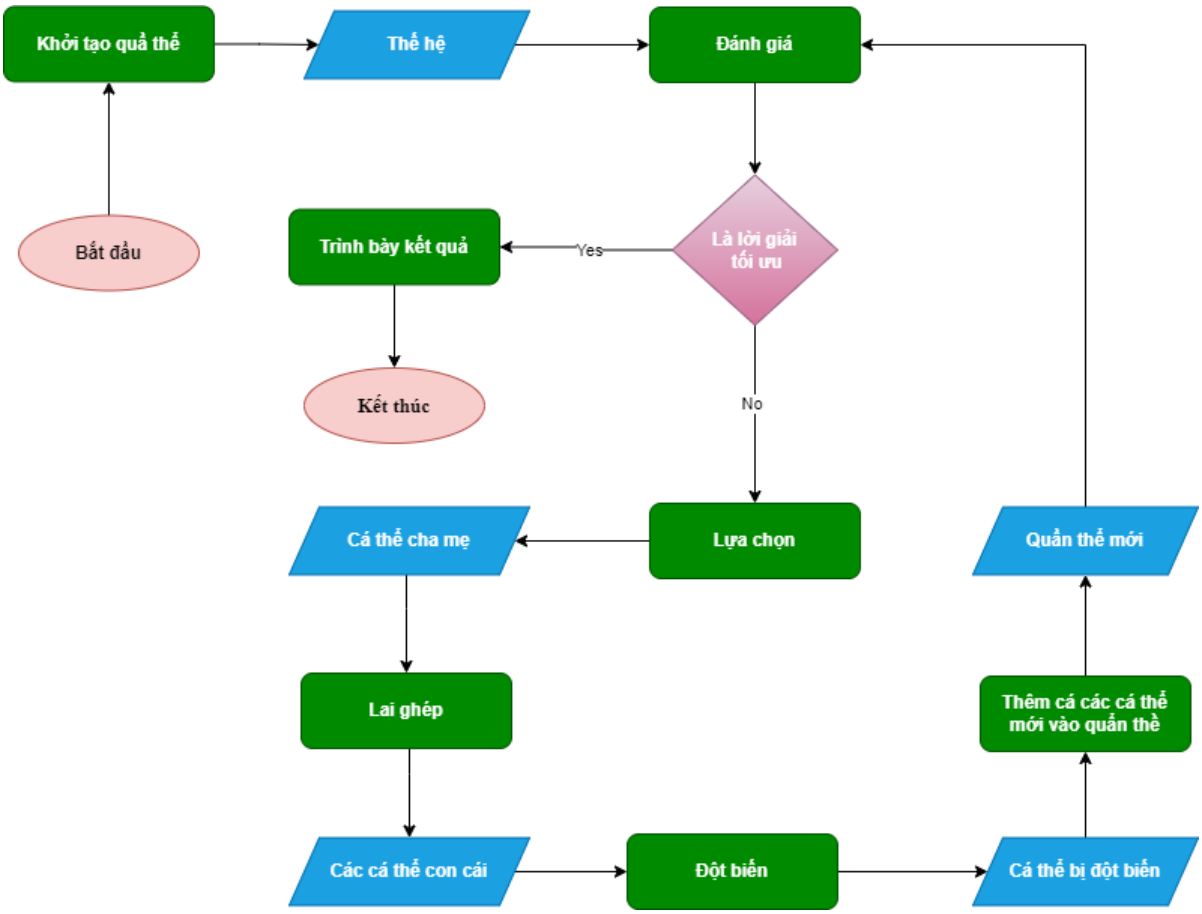
e) Đột biến (Mutation)

Thay đổi ngẫu nhiên một số phần trong NST để tạo ra sự đa dạng trong quần thể. Quá trình này mô phỏng lại quá trình đột biến gen trong tự nhiên

f) Lặp lại

Lặp lại các bước từ bước "**b**" đến bước "**e**" cho đến khi tìm được giải pháp tối ưu hoặc đạt đến số lượng thế hệ tối đa.

2.2.3. Sơ đồ của thuật giải



Hình 1. Sơ đồ của thuật giải di truyền (GA Flowchart)

Diễn giải:

Bước 1: Phát sinh quần thể ban đầu (Initialization):

Dựa vào kích thước quần thể (Population size) cho trước, phát sinh ngẫu nhiên ra số cá thể của quần thể ban đầu (Generation).

Bước 2: Xác định độ thích nghi của các cá thể trong quần thể ban đầu.

Tùy vào hàm mục tiêu (fitness) của bài toán cụ thể, ta sẽ có những hàm để đánh giá độ thích nghi cuả các cá thể.

Bước 3: Kiểm tra các cá thể ban đầu (hoặc sau một số lần lặp tái tạo quần thể mới), có cá thể nào đã đạt đến lời giải tối ưu hay chưa?

- Nếu tồn tại cá thể đạt đến lời giải tối ưu thì kết thúc và đến bước trình bày lời giải (Present the Solution).
- Nếu chưa thì tiếp tục qua bước chọn lọc (Selection).

Bước 4: Chọn lọc (Selection)

Lựa chọn các cá thể mạnh với độ thích nghi cao để tiến hành các thao tác sinh sản như lai ghép (Crossover) hoặc đột biến (Mutation) và loại bỏ đi các cá thể yếu với độ thích nghi thấp.

Bước 5: Lai ghép (Crossover)

Từ những cá thể được chọn lọc trước đó với độ thích nghi cao, ta tiến hành lai ghép các cá thể. Từ mỗi gen của 2 cá thể được chọn lọc từ thế hệ trước sẽ được phối hợp với nhau (theo một quy tắc nào đó) để tạo thành 2 gen mới.

Bước 6: Đột biến (Mutation)

Đó là sự biến đổi ngẫu nhiên một hoặc nhiều thành phần gen của một cá thể ở thế hệ trước tạo ra một cá thể hoàn toàn mới ở thế thệ sau. Nhưng thao tác này chỉ được phép xảy ra với tần suất rất thấp (thường dưới 0.01), vì thao tác này có thể gây xáo trộn và làm mất đi những cá thể đã chọn lọc và lai tạo có tính thích nghi cao, dẫn đến thuật toán không còn hiệu quả.

Bước 6: Lặp lại bước 2

Thế hệ mới được tạo ra lại được xử lý như thế hệ trước (xác định độ thích nghi và tạo thế hệ mới) cho đến khi có một cá thể đạt được lời giải mong muốn hoặc đạt đến thời gian giới hạn, thế hệ tối đa.

2.2.4. Nhận xét

a) Ưu điểm

- Tính đa dạng và khả năng tìm kiếm toàn cục: Thuật giải di truyền có khả năng duy trì và khám phá các vùng không gian giải pháp rộng lớn, giúp tìm ra giải pháp gần tối ưu trên không gian tìm kiếm.
- Khả năng làm việc trên không gian tìm kiếm lớn: Do tính chất đa dạng của quần thể, thuật giải di truyền có thể làm việc hiệu quả trên các vấn đề có không gian tìm kiếm lớn mà không gặp vấn đề về độ phức tạp tính toán.
- Khả năng tìm giải pháp gần tối ưu trong thời gian ngắn: Do khả năng làm việc song song trên nhiều cá thể trong quần thể, thuật giải di truyền có thể đạt được giải pháp gần tối ưu trong thời gian ngắn.
- Tính linh hoạt và dễ dàng thích ứng: Thuật giải di truyền có thể dễ dàng thích ứng với nhiều loại vấn đề khác nhau thông qua việc điều chỉnh các tham số như kích thước quần thể, xác suất lai ghép và đột biến.

b) Hạn chế

- Rủi ro rơi vào tình trạng hội tụ sớm (premature convergence): Thuật giải di truyền có thể dễ rơi vào tình trạng hội tụ sớm nếu không có cơ chế đảm bảo tính đa dạng của quần thể, dẫn đến việc bỏ qua các giải pháp tiềm ẩn tốt hơn.
- Phụ thuộc vào tham số cố định: Hiệu suất của thuật giải di truyền phụ thuộc nhiều vào việc lựa chọn các tham số như kích thước quần thể, xác suất lai ghép và đột biến. Việc chọn các giá trị thích hợp cho các tham số này không phải lúc nào cũng dễ dàng.
- Khả năng gây kẹt cục: Trong một số trường hợp, thuật giải di truyền có thể gây kẹt cục tại các điểm cục bộ, không thể thoát ra khỏi chúng để tìm kiếm giải pháp tối ưu toàn cục.

- Không hiệu quả trên các vấn đề có ràng buộc cứng: Thuật giải di truyền không hiệu quả trên các vấn đề có ràng buộc cứng hoặc các vấn đề yêu cầu đạo hàm để tối ưu.

c) Ứng dụng

- Tối ưu hóa: GA được sử dụng để tối ưu hóa các thiết kế, quy trình sản xuất, hệ thống logistics, v.v.
- Học máy: GA được sử dụng để huấn luyện các mô hình học máy, như mạng nơ-ron nhân tạo và máy hỗ trợ vector.
- Tài chính: GA được sử dụng để phân tích thị trường tài chính, phát triển các chiến lược đầu tư và quản lý rủi ro.
- Khoa học kỹ thuật: GA được sử dụng để giải quyết các bài toán trong nhiều lĩnh vực khoa học kỹ thuật, như thiết kế chip điện tử, phát triển thuốc và vật liệu mới.
- Tìm kiếm và tối ưu hóa cấu trúc hóa học: Trong hóa học, thuật giải di truyền có thể được sử dụng để tìm kiếm và tối ưu hóa cấu trúc phân tử hoá học với các tính chất mong muốn.
- Quy hoạch Lịch Trình: Trong quản lý dự án và sản xuất, thuật giải di truyền có thể được áp dụng để tối ưu hóa lịch trình sản xuất và quản lý tài nguyên.
- Phân tích dữ liệu và khai thác dữ liệu: Trong khoa học dữ liệu và khai thác dữ liệu, thuật giải di truyền có thể được sử dụng để khai thác các mẫu tiềm ẩn và xu hướng từ dữ liệu lớn.

2.3. Khai phá dữ liệu (Data Mining)

2.3.1. Khái niệm

Khai phá dữ liệu là quá trình khám phá và phân tích tự động các mẫu tiềm ẩn, thông tin và tri thức từ các bộ dữ liệu lớn. Mục tiêu của khai phá dữ liệu là tìm ra các mẫu, quy luật, và thông tin tiềm ẩn có giá trị từ dữ liệu đó để hỗ trợ quyết định hoặc dự đoán trong tương lai [4].

2.3.2. Mục đích của việc khai phá dữ liệu

- Phát hiện mẫu và quy luật tiềm ẩn: Khai phá dữ liệu giúp phát hiện ra các mẫu và quy luật tiềm ẩn trong dữ liệu, nhưng không thể được phát hiện bằng các phương pháp truyền thống. Điều này có thể bao gồm việc tìm ra mối quan hệ, xu hướng hoặc sự tương tác giữa các biến.
- Dự đoán và phân loại: Khai phá dữ liệu cung cấp các công cụ để dự đoán các biến mục tiêu hoặc phân loại các bản ghi vào các nhóm khác nhau dựa trên các đặc điểm của chúng. Điều này có thể giúp trong việc dự đoán xu hướng thị trường, xác định khách hàng tiềm năng hoặc phân loại dữ liệu y tế.
- Xác định thông tin quan trọng: Khai phá dữ liệu giúp xác định thông tin quan trọng từ dữ liệu lớn, từ đó giúp tập trung vào các khía cạnh quan trọng và loại bỏ thông tin không cần thiết hoặc không quan trọng.
- Hỗ trợ quyết định: Thông qua việc phân tích và tìm kiếm thông tin từ dữ liệu, khai phá dữ liệu cung cấp cơ sở cho quyết định thông minh và hiệu quả trong các lĩnh vực như kinh doanh, y tế, tài chính và hợp pháp.
- Tạo ra giá trị từ dữ liệu: Bằng cách tận dụng thông tin tiềm ẩn từ dữ liệu, khai phá dữ liệu giúp tổ chức tạo ra giá trị từ dữ liệu của mình thông qua việc tối ưu hóa hoạt động, tăng cường hiệu suất và cải thiện dịch vụ.

2.3.3. Ứng dụng của việc khai phá dữ liệu

- Kinh doanh và Marketing:
  - Xác định hành vi của khách hàng và dự đoán xu hướng mua sắm.
  - Phân loại khách hàng và tạo các chiến lược tiếp thị được cá nhân hóa.
  - Phát hiện gian lận trong giao dịch tài chính và giao dịch thẻ tín dụng.
- Y tế và Dược phẩm:
  - Dự đoán nguy cơ bệnh và phát hiện bất thường trong dữ liệu y tế.
  - Hỗ trợ chuẩn đoán bệnh và lên kế hoạch điều trị dựa trên lịch sử bệnh nhân và dữ liệu y khoa.
  - Tìm kiếm và phát triển các loại thuốc mới.
- Tài chính:
  - Dự đoán xu hướng thị trường và biến động giá cả.

- Phát hiện gian lận trong giao dịch tài chính và bảo vệ chống rủi ro.
- Tạo ra mô hình đánh giá rủi ro và quản lý danh mục đầu tư.

- Công nghệ thông tin và Internet:
  - Phân loại và lọc email rác (spam) trong hộp thư đến.
  - Tạo ra hệ thống khuyến nghị cho các dịch vụ trực tuyến như mua sắm và xem phim.
  - Phát hiện và ngăn chặn các cuộc tấn công mạng và các hành vi độc hại.

2.4. Khai phá luật kết hợp (Association Rule Mining)

2.4.1. Khái niệm

Khai phá luật kết hợp là quá trình tìm kiếm và trích xuất các quy tắc kết hợp giữa các mục trong dữ liệu.

2.4.2. Mục tiêu của khai phá luật kết hợp

Tìm tần số mẫu, mối kết hợp, sự tương quan hay các cấu trúc nhân quả giữa các tập đối tượng trong các CSDL giao tác, CSDL quan hệ, và những kho thông tin khác để giúp hiểu rõ hơn về các mối liên kết và hành vi của dữ liệu.

Việc áp dụng hiệu quả khai phá luật kết hợp có thể giúp các tổ chức nâng cao hiệu quả hoạt động, tăng doanh thu và lợi nhuận.

2.4.3. Các định nghĩa trong khai phá luật kết hợp

a) Độ hỗ trợ (Support)

Độ hỗ trợ được định nghĩa là các hạng mục được tìm thấy với tần suất tối thiểu trong toàn bộ tập dữ liệu. Độ hỗ trợ (hoặc tần suất xuất hiện) của một mẫu A, trong đó A là một tập con của I, là số lượng giao dịch chứa A trong DB. Công thức tính độ support như sau: $\mathbf{support(A)} = \frac{\boldsymbol{count(A)}}{\boldsymbol{|DB|}}$

Trong đó: count(A) là số lần các giao dịch có chứa A trong DB.

b) Tập phổ biến (frequent itemsets)

Tập phổ biến là tập có độ phổ biến thỏa mãn độ phổ biến tối thiểu (minSup).

Một mẫu A được coi là phổ biến nếu độ hỗ trợ của A không nhỏ hơn một ngưỡng hỗ trợ tối thiểu minSup được xác định trước.

Nếu support(A) ≥ minSup thì A là tập phổ biến.

Tính chất tập phổ biến: Tất cả các tập con của mẫu phổ biến đều là mẫu phổ biến.

Nếu độ hỗ trợ tối thiểu cao thì sẽ dẫn đến kết quả là ít tập phần tử (itemset) phổ biến dẫn bến các tập dữ liệu trên rất thường xuyên (có độ support cao).

Ngược lại, độ hỗ trợ tối thiểu thấp sẽ dẫn đến xuất hiện nhiều tập phần tử phổ biến hiếm, ít phổ biến xuất hiện (support cao thấp).

c) Luật kết hợp

Sự kết hợp: Các phần tử cùng xuất hiện với nhau trong một hay nhiều giao dịch. Thể hiện mối liên hệ giữa các phần tử / các tập phần tử.

Qui tắc kết hợp có điều kiện giữa các tập phần tử. Thể hiện mối liên hệ (có điều kiện) giữa các phần tử. Cho A và B là các tập phần tử, luật kết hợp giữa A và B là AB dẫn đến B xuất hiện trong điều kiện A xuất hiện.

Một luật kết hợp là một câu lệnh có điều kiện nói rằng, nếu hạng mục A tồn tại trong giao dịch thì có khả năng hạng mục B cũng có trong giao dịch đó.

Một mẫu A = {X, Y}, ta có luật kết hợp $X \rightarrow Y$. Độ tin cậy được định nghĩa là tần suất của một hạng mục liên quan đến tập mục chứa các hạng mục được hỗ trợ. Độ tin cậy (confidence) là xác suất xảy ra Y khi đã biết X.

Công thức được tính như sau: $\mathbf{confidence(X \rightarrow Y) = P(Y \mid X)} = \frac{\boldsymbol{count(X \cup Y)}}{\boldsymbol{count(X)}}$

Trong đó: count($X \cup Y$) là số giao dịch chứa ($X \cup Y$) và count(X) là số giao dịch chứa X.

2.4.4. Một số thuật toán khai phá luật kết hợp

- Apriori (1994): Tìm kiếm theo chiều rộng.
- Paritition (1995): Tương tự Apriori, dùng phần giao tập hợp để xác định giá trị support.
- Eclat (1997): Kết hợp duyệt chiều sâu và phần giao tidlist.
- FP-Growth (2000): Duyệt cây phát triển mẫu theo chiều sâu.

2.4.5. Hạn chế của khai thác luật kết hợp so với khai thác tập hữu ích cao

- Thiếu thông tin về mức độ quan trọng của luật: Tập phổ biến chỉ cung cấp thông tin về sự phổ biến của các mẫu itemset, nhưng không đánh giá mức độ quan trọng hoặc ý nghĩa của các luật. Điều này có thể dẫn đến việc bỏ lỡ các luật quan trọng nhưng có tần suất xuất hiện thấp. Ví dụ: Trong một cửa hàng bán lẻ, một mẫu itemset "bột giặt" và "nước rửa chén" có thể xuất hiện phổ biến, nhưng không đảm bảo rằng việc mua hai sản phẩm này cùng một lúc là quan trọng hoặc có ý nghĩa đối với khách hàng.
- Không hiệu quả trong việc loại bỏ luật không hữu ích: Tập phổ biến có thể chứa nhiều luật không hữu ích hoặc không mong muốn, nhưng không cung cấp thông tin đủ để loại bỏ chúng một cách hiệu quả. Điều này có thể gây ra lãng phí tài nguyên khi phân tích và sử dụng các luật. Ví dụ: Một tập dữ liệu giao dịch có thể chứa nhiều luật không mong muốn như "mua bánh mì => mua ô-liu". Mặc dù luật này có thể xuất hiện trong tập phổ biến, nhưng không có ý nghĩa và không nên được sử dụng trong quyết định kinh doanh.
- Khả năng phân tích hạn chế về mối quan hệ giữa các biến: Tập phổ biến tập trung vào sự xuất hiện chung của các mẫu itemset mà không cung cấp thông tin về mối quan hệ giữa các biến. Điều này có thể làm giảm khả năng hiểu rõ về cơ chế hoặc mối quan hệ trong dữ liệu. Ví dụ: Một tập dữ liệu khách hàng của một trang web mua sắm có thể chỉ cho thấy rằng một số mẫu sản phẩm được mua cùng nhau một cách phổ biến, nhưng không cung cấp thông tin về lý do hoặc mối quan hệ giữa các sản phẩm đó.
- Không đánh giá được tính tin cậy của luật: Tập phổ biến không đánh giá được tính tin cậy hoặc ý nghĩa của các luật. Điều này làm cho việc xác định sự đáng tin cậy của các luật và đánh giá khả năng áp dụng của chúng trở nên khó khăn. Ví dụ: Một tập dữ liệu chứa thông tin về các giao dịch tín dụng có thể chỉ cho thấy rằng một số mẫu giao dịch phổ biến, nhưng không đảm bảo rằng các luật kết hợp xuất hiện trong tập phổ biến cũng có độ tin cậy cao.
- Không tận dụng được thông tin chi tiết về mối quan hệ giữa biến: Tập phổ biến không cung cấp thông tin chi tiết về mối quan hệ giữa các biến trong các luật kết hợp. Điều này có thể làm mất đi cơ hội tận dụng thông tin chi tiết để hiểu rõ

hơn về dữ liệu và tạo ra các luật kết hợp có ý nghĩa. Ví dụ: Một tập dữ liệu về hành vi truy cập trang web có thể chỉ cho thấy rằng một số trang web thường được truy cập cùng nhau, nhưng không cung cấp thông tin về cách mà các trang web này liên quan đến nhau.

2.4.6. Ứng dụng

- Phân tích dữ liệu giỏ hàng (basket data analysis), thương mại, ...
    - Hiểu rõ hành vi mua sắm của khách hàng: Phân tích các mặt hàng thường được mua cùng nhau giúp doanh nghiệp hiểu rõ hơn về sở thích và nhu cầu mua sắm của khách hàng. Ví dụ: quy tắc kết hợp bia - snack, sữa chua - ngũ cốc, v.v.
    - Đề xuất sản phẩm phù hợp: Dựa trên các quy tắc kết hợp, doanh nghiệp có thể đề xuất các sản phẩm liên quan đến sản phẩm mà khách hàng đang xem hoặc đã mua. Ví dụ: khi khách hàng mua sữa chua, hệ thống có thể đề xuất thêm ngũ cốc, trái cây, v.v.
    - Tối ưu hóa bố trí sản phẩm: Doanh nghiệp có thể sắp xếp các sản phẩm thường được mua cùng nhau trên kệ hàng hoặc trang web để thu hút sự chú ý của khách hàng và tăng khả năng mua hàng.
    - Phát triển chiến lược khuyến mãi hiệu quả: Doanh nghiệp có thể tạo ra các chương trình khuyến mãi kết hợp các sản phẩm thường được mua cùng nhau để kích thích nhu cầu mua sắm của khách hàng. Ví dụ: mua bia tặng snack, mua sữa chua tặng ngũ cốc, v.v.
- Tiếp thị chéo (cross-marketing).
    - Xác định khách hàng tiềm năng: Phân tích dữ liệu giỏ hàng giúp doanh nghiệp xác định những khách hàng có khả năng mua thêm các sản phẩm khác. Ví dụ: khách hàng mua sữa chua có thể có tiềm năng mua thêm ngũ cốc, trái cây, v.v.
    - Gửi chiến dịch tiếp thị phù hợp: Doanh nghiệp có thể gửi các chiến dịch tiếp thị qua email, tin nhắn SMS, quảng cáo trực tuyến, v.v. để giới thiệu các sản phẩm liên quan đến sản phẩm mà khách hàng đã mua hoặc có khả năng mua.

  - Tăng tỷ lệ chuyển đổi: Tiếp thị chéo dựa trên dữ liệu giỏ hàng có thể giúp doanh nghiệp tăng tỷ lệ chuyển đổi khách hàng tiềm năng thành khách hàng mua hàng.
- Phân loại dữ liệu (classification) và gom cụm dữ liệu (clustering) với các mẫu phổ biến.
  - Phân loại khách hàng: Doanh nghiệp có thể phân loại khách hàng dựa trên hành vi mua sắm của họ. Ví dụ: khách hàng mua nhiều sữa chua có thể được phân loại vào nhóm khách hàng yêu thích đồ ăn sáng, khách hàng mua nhiều bia có thể được phân loại vào nhóm khách hàng yêu thích tụ tập bạn bè, v.v.
  - Phân tích xu hướng thị trường: Phân tích dữ liệu giỏ hàng theo thời gian có thể giúp doanh nghiệp xác định xu hướng thị trường và đưa ra các quyết định kinh doanh phù hợp. Ví dụ: doanh nghiệp có thể nhận thấy nhu cầu mua sắm trực tuyến tăng cao trong thời gian dịch bệnh và điều chỉnh chiến lược kinh doanh cho phù hợp.
- Thiết kế catalogue, Website, đồ họa, ...
  - Sắp xếp sản phẩm hợp lý: Dữ liệu giỏ hàng có thể được sử dụng để sắp xếp các sản phẩm trong catalogue hoặc trên website theo cách mà khách hàng thường mua cùng nhau. Ví dụ: các sản phẩm thường được mua cùng nhau có thể được đặt cạnh nhau trên cùng một trang web hoặc trong cùng một danh mục.
  - Thiết kế giao diện thu hút: Doanh nghiệp có thể sử dụng dữ liệu giỏ hàng để thiết kế giao diện website hoặc catalogue thu hút sự chú ý của khách hàng và khuyến khích họ mua hàng. Ví dụ: doanh nghiệp có thể sử dụng hình ảnh bắt mắt cho các sản phẩm thường được mua cùng nhau hoặc hiển thị các khuyến mãi cho các sản phẩm kết hợp.
  - Cá nhân hóa trải nghiệm khách hàng: Dữ liệu giỏ hàng có thể được sử dụng để cá nhân hóa trải nghiệm khách hàng trên website hoặc trong ứng dụng di động. Ví dụ: hệ thống có thể đề xuất các sản phẩm phù hợp với sở thích mua sắm của từng khách hàng.

2.5. Khai thác tập mục hữu ích cao (High utility itemset mining)

2.5.1. Giới thiệu về tập mục hữu ích cao

Khai thác tập mục phổ biến (Frequent Itemset Mining - FIM) là một loại khai thác dữ liệu phổ biến và cần thiết đến một loạt các ứng dụng. Với tập dữ liệu giao dịch, FIM bao gồm phát hiện các tập mục thường xuyên. Tức là các nhóm tập mục (itemset) xuất hiện thường xuyên trong giao dịch. Tuy nhiên, một hạn chế quan trọng của FIM là nó giả định rằng mỗi tập mục không thể xuất hiện quá một lần trong một giao tác (có hoặc không có) và tất cả các mặt hàng đều có tầm quan trọng như nhau (trọng lượng, đơn vị lợi nhuận, giá trị, …) [2]. Nhưng giả định này thường không đúng trong các ứng dụng thực tế. Để giải quyết vấn đề này, vấn đề của FIM đã được xác định lại bởi HUIM [1].

Khai thác tập mục hữu ích cao (High Utility Itemset Mining - HUIM) là một nhánh con của khai phá dữ liệu tập trung vào việc tìm kiếm các tập con của các phần tử trong tập dữ liệu có giá trị cao. Khác với khai phá dữ liệu truyền thống, tập trung vào việc tìm kiếm các tập con có tần suất xuất hiện cao (tập phổ biến), HUIM quan tâm đến cả tần suất xuất hiện và giá trị của các tập con.

Giá trị của một tập con được đo bằng một hàm hữu ích, ví dụ như lợi nhuận, doanh thu, hoặc số lượng khách hàng. Mục tiêu của HUIM là tìm ra các tập con có giá trị cao nhất, đồng thời đảm bảo rằng các tập con này có tần suất xuất hiện tối thiểu [3].

Khái quát hóa của khai thác tập mục hữu ích cao:

- Các item có thể xuất hiện nhiều lần trong một giao dịch. Ví dụ như một khách hàng có thể mua hai chai sữa trong một giao dịch.
- Các item có một đơn vị hữu ích. Ví dụ một chai sữa có đơn giá là 20.000 thể đem lại giá trị lợi nhuận là 12.000 VNĐ (hữu ích ở trường hợp này là lợi nhuận từ chai sữa).

2.5.2. Mục tiêu của khai thác tập mục hữu ích cao

Mục tiêu chính của khai thác tập mục hữu ích cao (High utility itemset mining) là tìm ra các tập hợp các mặt hàng (itemset) mà khi kết hợp với nhau sẽ tạo ra giá trị lợi ích cao nhất. Trong ngữ cảnh này, "lợi ích" thường được hiểu là giá trị tổng hợp hoặc tổng số lượng, thay vì chỉ số hỗ trợ như trong khai thác tập phổ biến.

Ví dụ: Giả sử chúng ta có một tập dữ liệu ghi lại các giao dịch mua hàng tại một cửa hàng tạp hóa. Tập dữ liệu này bao gồm các thông tin về các sản phẩm được mua trong mỗi giao dịch và số lượng mua của mỗi sản phẩm.

Hàm hữu ích trong trường hợp này có thể là lợi nhuận. Lợi nhuận của một tập con là tổng số lượng mua của mặt hàng nhân với đơn giá của mặt hàng.

Mục tiêu của HUIM trong trường hợp này là tìm ra các tập con sản phẩm có lợi nhuận cao nhất.

2.5.3. Các định nghĩa trong khai thác tập hữu ích cao

Cho $I = \{i_1, i_2, ..., i_m\}$ là một tập các item riêng biệt. Một giao tác $T_j = \{x_l \mid l = 1, 2, ..., N_j, x_l \in I\}$, trong đó N_j là số item trong giao tác T_j. Một CSDL giao tác D chứa các giao tác, $DB = \{T1, T2, ..., Tn\}$, trong đó n là tổng số các giao tác trong CSDL [3].

Bảng 1. CSDL giao tác trong khai thác tập mục hữu ích cao

TID (D)	Items (I)	Hữu ích nội (IU)	Độ hữu ích (U)	Hưu ích giao tác (TU)
T1	a, c, d	1, 1, 1	5, 1, 2	8
T2	a, c, e, g	2, 6, 2, 5	10, 6, 6, 5	27
T3	a, b, c, d, e, f	1, 2, 1, 6, 1, 5	5, 4, 1, 12, 3, 5	30
T4	b, c, d, e	4, 3, 3, 1	8, 3, 6, 3	20
T5	b, c, e, g	2, 2, 1, 2	4, 2, 3, 2	11
T6	a, c, d	3, 3, 3	15, 3, 6	24
T7	a, b, c, d, f	1, 1, 1, 2, 3	5, 2, 1, 4, 3	15
T8	a, b, c, e, f	1, 2, 2, 1, 1	5, 4, 2, 3, 1	15
Tổng hữu ích các giao tác				**150**

Bảng 2. Giá trị hữu ích của các item

Item	Hưu ích ngoại (EU)
A	5
B	2
C	1
D	2
E	3
F	1
G	1

a) Hữu ích ngoại (EU) của một item

Hữu ích ngoại của item $i_x \in I$ được ký hiệu là $EU(i_x)$, là giá trị hữu ích của item i_x trong bảng giá trị hữu ích của các item (bảng 3) [3] [4].

Ví dụ (a): Từ bảng 3, ta có item b có giá trị hữu ích ngoại là 2 => EU(b) = 2.

b) Hữu ích nội (IU) của một item trong một giao tác (T)

Hữu ích nội của item $i_x \in I$ trong giao dịch $T_y \in DB$, ký hiệu là $IU(i_x, T_y)$, là giá trị đếm gắn với i trong T trong bảng giao dịch của DB [3] [4] [5].

Ví dụ (b): Từ bảng 2, ta có item b có độ hữu ích nội là 2 trong giao tác T3 => IU(b, T3) = 2.

c) Hữu ích của một item trong một giao tác

Hữu ích của mục $i_x \in I$ trong giao dịch $T_y \in DB$, ký hiệu là $U(i_x, T_y)$ được tính bằng tích của $IU(i_x, T_y)$ và $EU(i_x)$, trong đó $U(i_x, T_y) = IU(i_x, T_y) \times EU(i_x, T_y)$. [3]

Ví dụ (c): Ta có EU(b) từ ví dụ (a) và $IU(b, T_3)$ từ ví dụ (b) => hữu ích của item b trong giao tác T3 được tính như sau: $U(b, T_3) = IU(b, T_3) \times EU(b) \Leftrightarrow U(b, T_3) = 2 \times 2 = 4$.

d) Hữu ích của một itemset trong một giao tác

Hữu ích của một itemset $\{i_x, i_y\}$ gọi là X trong giao dịch $T \in DB$, ký hiệu là U(X, T) hoặc $U(i_x, T).U(i_y, T)$, là tổng hữu ích của tất cả các item chứa trong X và nằm trong giao dịch T mà trong đó X được chứa, trong đó [3] [4]:

$$U(X, T) = \sum_{i \in X \wedge X \subseteq T} U(i, T)$$

Ví dụ (d): Từ T1 trong bảng 2, ta có giá trị hữu ích của item a là 5, và item c là 1. Ta có thể kết luận giá hữu ích của itemset {a, c} $\Leftrightarrow$ U({a,c}, T1) = U(a, T1) + U(c, T1) = 5 +1 = 6.

e) Hữu ích của một itemset

Hữu ích của tập mục X, ký hiệu là U(X), là tổng hữu ích của X trong tất cả các giao tác có chứa X trong DB, trong đó [4] [5]:

$$U(X) = \sum_{T \in DB \wedge X \subseteq T} U(X, T)$$

Ví dụ (e): Từ bảng 2, itemset {a, b} xuất hiện trong các giao dịch T3, T7 và T8 với các hữu ích tương ứng mỗi giao tác là U({a, b}, T_3), U({a, b}, T_7), U({a, b}, T_8). Ta có thể kết luận:

$$U(\{a, b\}) = U(\{a, b\}, T_3) + U(\{a, b\}, T_7) + U(\{a, b\}, T_8)$$

$$\Leftrightarrow (5 + 4) + (5 + 2) + (5 + 4) = 25.$$

f) Hữu ích giao tác (TU)

Hữu ích của giao tác T, ký hiệu là TU(T), là tổng các hữu ích của tất cả các item chứa trong T, trong đó [3] [4]:

$$TU(T) = \sum_{i \in T} U(i, T)$$

Ví dụ (f): Trong giao tác T1 từ bảng 2 chứa các item gồm {a, c, d}

$$\Rightarrow TU(T_1) = U(a, T_1) + U(c, T_1) + U(d, T_1) = 5 + 1 + 2 = 8$$

g) Trọng số hữu ích của giao tác

Trọng số hữu ích giao tác của tập mục X trong DB, ký hiệu là twu(X), là tổng hữu ích của tất cả các giao dịch có chứa X trong DB, trong đó [3] [4]:

$$TWU(X) = \sum_{T \in DB \wedge X \in T} TU(T)$$

Ví dụ (g):

- Ta muốn tìm TWU({a, c}), từ bảng 1 ta thấy rằng các giao dịch chứa item set{a, c} gồm T1, T2, T6.
- Từ ví dụ (f) ta có $TU(T_1) = 8$.
- Tiếp theo $TU(T_2) = U(a, T_2) + U(c, T_2) + U(e, T_2) + U(g, T_2) = 10 + 6 + 6 + 5 = 27$.
- Tiếp theo $TU(T_6) = U(a, T_6) + U(c, T_6) + U(d, T_6) = 15 + 3 + 6 = 23$.
- => $TWU(\{a, c\}) = TU(T_1) + TU(T_2)\ TU(T_6) = 8 + 27 + 23 = 58$.

Tính chất 1: Nếu TWU(X) nhỏ hơn một ngưỡng giá trị tối thiểu (minutil) nhất định thì tất cả các siêu tập hợp (X mở rộng) của X đều không có tính hữu ích cao [2] [5].

Giả sử chúng ta đang khai thác các tập mục hữu ích cao với ngưỡng tối thiểu cho trước là 60. Thì bất cứ itemset mở rộng nào từ {a, c} đều không có tính hữu ích cao, vì chúng

không bao giờ lớn hơn 58 (Tính chất trên tôn trọng tính chất chống đơn điệu từ thuật toán Apriori).

Tính chất 2: Cho X ⊂ Y, ∀X, Y ⊂ I thì trọng số hữu ích của tập mục X luôn luôn lớn hơn hoặc bằng trọng số hữu ích của tập mục Y. Như vậy, trọng số hữu ích giao tác của tập mục thỏa mãn tính chất bao đóng giảm dần [2].

Tính chất 3: Cho X là một tập mục, nếu TWU(X) < minUtility thì tập mục X và tất cả tập cha của X không phải là tập mục hữu ích cao.

Bảng 3. Trọng số hữu ích của giao tác trong bảng 1

Item	g	f	b	d	e	a	c
TWU	38	60	91	97	103	119	120

h) Hữu ích còn lại (RU).

Hữu ích còn lại là thước đo giá trị tiềm năng có thể tìm thấy bằng cách kết hợp một tập hợp con với các mục khác trong tập dữ liệu. Nó đại diện cho số lượng hữu ích bổ sung tối đa có thể đạt được bằng cách mở rộng tập hợp con trong một giao tác, trong đó [3] [4]:

$$RU(X, T) = \sum_{x \subseteq T \in D} RU(X, T)$$

Ví dụ (h): Trong giao tác T2 từ bảng 2, với X = {a, c} thì phần còn lại của X trong giao tác T2 sẽ là {e, g} ⇔ $RU(\{a, c\}, T_2) = U(\{e, g\}, T_2)$ ⇔ $RU(\{a, c\}, T_2) = 6 + 5 = 11$.

i) Tập mục hữu ích cao (HUI)

Một tập mục X được gọi là tập mục hữu ích cao trong CSDL D, nếu giá trị hữu ích của X không nhỏ hơn ngưỡng hữu ích tối thiểu (được đưa ra bởi người dùng). Gọi HUIs là tập các tập mục hữu ích cao thì [2]:

$$HUIs = \{X | X \in I, U(X) >= minUtility\}$$

Ví dụ (i): Với minUtility = 40, thì tập mục hữu ích cao được khai thác từ bảng 1 và bảng 2 gồm các tập mục ở bảng 4.

Bảng 4. Các tập mục hữu ích cao

STT	Itemset	Utility
1	{a}	45
2	{a, c}	59
3	{a, d}	54
4	{a, c, d}	60
5	{a, c, e}	41
6	{b, c, d}	41
7	{b, c, e}	40
8	{a, b, d, f}	40
9	{b, c, d, e}	40
10	{a, b, c, d, f}	42

2.5.4. Nguyên lý hoạt động của một số thuật toán khai thác tập mục hữu ích cao

Các thuật toán như Two – Phase (PAKDD 2005) và UPGrowth (KDD 2010) hoạt động như sau [5]:

- Giai đoạn 1: Tìm từng tập mục X sao cho TWU(X) ≥ minutil bằng cách sử dụng giới hạn trên của TWU để cắt bớt không gian tìm kiếm.
- Giai đoạn 2: Cơ sở dữ liệu có thể tính toán lại hữu ích chính xác của các tập mục còn lại. Và trả về các tập mục hữu ích cao.

Ngoài ra còn một số các thuật toán cải tiến về mặt thời gian và bộ nhớ khai thác tập mục hữu ích cao như:

- HUI – Miner
- FHM
- …

2.5.5. Một số hạn chế của việc khai thác tập mục hữu ích cao

Việc khai thác tập mục có độ hữu ích cao vẫn là một nhiệm vụ rất tốn kém về mặt thời gian và bộ nhớ:

- Thuật toán HUIM có thể tốn nhiều thời gian và bộ nhớ khi xử lý các tập dữ liệu lớn.
- Việc tính toán giá trị hữu ích và hữu ích còn lại cho các tập con có thể tốn kém về mặt tính toán, đặc biệt là khi tập dữ liệu có mật độ cao và nhiều item.
- Việc tìm kiếm các tập con có giá trị cao có thể đòi hỏi nhiều lần lặp và khám phá, dẫn đến tốn thời gian và tài nguyên tính toán.
- Việc lựa chọn hàm hữu ích không phù hợp có thể dẫn đến việc tìm ra các tập con không có ý nghĩa thực tế hoặc không mang lại lợi ích cao cho ứng dụng.
- Khó khăn trong việc xử lý các tập dữ liệu thưa thớt.
- Khó khăn trong việc xử lý các tập dữ liệu có nhiều cấp độ.

2.5.6. Ưu điểm

- Tập trung vào việc tìm kiếm các tập con có giá trị cao, thay vì chỉ tập trung vào tần suất xuất hiện như khai phá dữ liệu truyền thống.Ví dụ trong phân tích dữ liệu giỏ hàng, HUIM có thể giúp tìm ra các tập hợp sản phẩm thường được mua cùng nhau và mang lại lợi nhuận cao cho cửa hàng.
- Có thể được áp dụng cho nhiều loại dữ liệu khác nhau, bao gồm dữ liệu giao dịch, dữ liệu khách hàng, dữ liệu y tế, ...Ví dụ như phân tích dữ liệu y tế để chẩn đoán bệnh và đề xuất phương pháp điều trị hiệu quả hơn.
- Có thể được kết hợp với các kỹ thuật khai phá dữ liệu khác để tăng cường hiệu quả và khả năng áp dụng. Ví dụ như kết hợp với kỹ thuật phân cụm để phân chia dữ liệu thành các nhóm có cùng đặc điểm, sau đó áp dụng HUIM cho từng nhóm để tìm ra các tập con hữu ích cao trong từng nhóm.
- Xử lý các tập dữ liệu phức tạp có nhiều item và nhiều cấp độ, áp dụng cho các ứng dụng thực tế phức tạp hơn như dữ liệu bán hàng đa kênh, bao gồm dữ liệu bán hàng trực tuyến, dữ liệu bán hàng qua cửa hàng, …

2.6. Thuật toán HUI - Miner

2.6.1. Giới thiệu

HUI-Miner (viết tắt của High-Utility Itemset Miner) là một thuật toán khai thác dữ liệu được phát triển để tìm kiếm các Tập Mục Lợi Ích Cao **(**HUI**)** trong cơ sở dữ liệu giao dịch. Thuật toán này được giới thiệu lần đầu tiên vào năm 2012 bởi Liu và Qu trong bài báo khoa học mang tên "Mining High-Utility Itemsets with Utility-Lists".

2.6.2. Nguyên tắc hoạt động

HUI-Miner sử dụng một cấu trúc dữ liệu mới gọi là danh sách lợi ích **(**utility-list**)** để lưu trữ thông tin về các sản phẩm và mức độ lợi ích của chúng trong mỗi giao dịch. Cấu trúc này giúp thuật toán có thể truy cập và xử lý dữ liệu hiệu quả hơn, từ đó tăng tốc độ khai thác HUI [2].

Cấu trúc utility-list:

Là một tập gồm các thành phần:

- TID: ID của giao dịch, có chứa tập mục.
- U: Giá trị hữu ích của tập mục.
- RU: Giá trị hữu ích còn lại của tập mục trong giao tác.

Cắt tỉa không gian tìm kiếm: Cho tập mục X, nếu tổng của U(X) và RU(X) nhỏ hơn ngưỡng hữu ích tối thiểu thì tập mục X và các phần mở rộng của tập mục X đều là tập mục hữu ích thấp.

Ví dụ ngưỡng tối thiểu là 20, hữu ích của tập mục {a} = 5 và hữu ích còn lại của a là 10 => U({a}) + RU({a}) = 5 + 10 = 15 < ngưỡng tối thiểu. Nên tập {a} và các tập mở rộng của a như {a,b}, {a, b, c}, {a,b, c, …} đều là tập mục hữu ích thấp.

Tính chất (tổng của U và RU): Giả sử có tập mục X và các tập mục mở rộng của X được tạo ra bằng cách thêm vào các tập mục Y. Nếu tổng giá trị U và RU trong utility-list < minUtility thì các thành phần mở rộng của X và các phần mở rộng bắc cầu của chúng đều là tập mục hữu ích thấp [3].

Quy trình hoạt động:

- Đọc dữ liệu giao dịch: Bước đầu tiên là đọc dữ liệu giao dịch từ cơ sở dữ liệu. Dữ liệu này thường bao gồm thông tin về các sản phẩm được mua trong mỗi giao dịch, cùng với giá trị lợi ích của mỗi sản phẩm.
- Tạo danh sách lợi ích: Dữ liệu giao dịch được sử dụng để tạo ra danh sách lợi ích cho mỗi sản phẩm. Danh sách lợi ích lưu trữ thông tin về các giao dịch có chứa sản phẩm và mức độ lợi ích của sản phẩm trong mỗi giao dịch.
- Tìm kiếm các tập mục ứng cử viên: Sử dụng danh sách lợi ích, HUI-Miner xác định các tập mục ứng cử viên có khả năng là HUI. Các tập mục ứng cử viên này được lựa chọn dựa trên mức độ lợi ích và sự hỗ trợ của chúng.
- Đánh giá và lọc các tập mục ứng cử viên: HUI-Miner đánh giá mức độ lợi ích của mỗi tập mục ứng cử viên và loại bỏ các tập mục không thỏa mãn ngưỡng lợi ích tối thiểu.
- Trả về kết quả: Cuối cùng, HUI-Miner trả về danh sách các tập mục được xác định là HUI.

2.6.3. Ưu điểm

- Hiệu quả: HUI-Miner sử dụng cấu trúc dữ liệu danh sách lợi ích giúp tăng tốc độ truy cập và xử lý dữ liệu, từ đó tăng tốc độ khai thác HUI.
- Độ chính xác cao: HUI-Miner đảm bảo tìm ra tất cả các HUI thỏa mãn ngưỡng lợi ích và hỗ trợ tối thiểu do người dùng đặt ra.
- Khả năng mở rộng: HUI-Miner có thể được áp dụng cho các tập dữ liệu có kích thước lớn mà không gặp vấn đề về hiệu suất.

2.7. Minh họa thuật toán

Cho tập dữ liệu hữu ích giao tác từ bảng 1 và bảng hữu ích của các mục từ bảng 2. Với minUtility là 40, ta khởi tạo danh sách các utility-list của các tập mục với cấu trúc utility ở định nghĩa trên:

- Với U bằng hữu ích của item tại giao tác (TID) đang xét.

  Ví dụ: $U(\{d\}, T_3) = 12$

- RU bằng tổng hữu ích còn lại của mục đó trên giao tác

  Ví dụ: $RU(\{d\}, T_3) = U(\{e\}, T_3) + U(\{a\}, T_3) + U(\{c\}, T_3) = 3 + 5 + 1 = 9$

Bảng 5. Sắp xếp các items từ bảng 1 theo TWU giảm dần từ bảng 3

TID	Items	U (Hữu ích)	TU (Hưu ích giao tác)
T1	d, a, c	2, 5, 1	8
T2	e, a, c	6, 10, 6	22
T3	f, b, d, e, a, c	5, 4, 12, 3, 5, 1	30
T4	b, d, e, c	8, 6, 3, 3	20
T5	b, e, c	4, 3, 2	9
T6	d, a, c	6, 15, 3	24
T7	f, b, d, a, c	3, 2, 4, 5, 1	15
T8	f, b, e, a, c	1, 4, 3, 5, 2	15

Khởi tạo cấu trúc utility-list với kích thước k = 1: Quét CSDL lần thứ nhất để tính TWU của các mục. Khi TWU của các mục đơn được tính, các mục có TWU nhỏ hơn minUtility sẽ được loại bỏ và thu được tập mục hữu ích.

- TWU(g) = 38 <= minUtility -> loại.
- Các TWU của các mục còn lại > minUtility -> Tạo utility-list.

a		
TID	U	RU
T1	5	1
T2	10	6
T3	5	1
T6	15	3
T7	5	1
T8	5	2
SUM	45	14

b		
TID	U	RU
T3	4	21
T4	8	12
T5	4	5
T7	2	10
T8	4	10
SUM	22	58

c		
TID	U	RU
T1	1	0
T2	6	0
T3	1	0
T4	3	0
T5	2	0
T6	3	0
T7	1	0
T8	2	0
SUM	19	0

d		
TID	U	RU
T1	2	6
T3	12	9
T4	6	6
T6	6	18
T7	4	6
SUM	30	45

e		
TID	U	RU
T2	6	16
T3	3	6
T4	3	3
T5	3	2
T8	3	7
SUM	18	34

f		
TID	U	RU
T3	5	25
T7	3	12
T8	1	14
SUM	9	51

Hình 2. Các utility-list có kích thước k = 1

Khởi tạo cấu trúc utility-list với kích thước k = 2 bằng phép nối: Để xây dựng utility-list của các tập mục gồm hai mục {ac} không cần phải quét CSDL mà chỉ cần thực hiện phép giao giữa utility-list của a và utility-list của c.

a, b		
TID	U	RU
T3	9	1
T7	7	1
T8	9	2
SUM	25	4

a, c		
TID	U	RU
T1	6	0
T2	16	0
T3	6	0
T6	18	0
T7	6	0
T8	7	0
SUM	59	0

a, d		
TID	U	RU
T1	7	1
T3	17	1
T6	21	1
T7	9	1
SUM	54	4

a, e		
TID	U	RU
T2	16	6
T3	8	1
T8	8	2
SUM	32	9

a, f		
TID	U	RU
T3	10	1
T7	8	1
T8	6	2
SUM	24	4

b, c		
TID	U	RU
T3	5	0
T4	11	0
T5	6	0
T7	3	0
T8	6	0
SUM	31	0

b, d		
TID	U	RU
T3	14	9
T4	14	6
T7	8	5
SUM	36	20

b, e		
TID	U	RU
T3	7	6
T4	11	3
T5	7	2
T8	7	7
SUM	32	18

b, f		
TID	U	RU
T3	9	21
T7	5	10
T8	5	10
SUM	19	41

c, d		
TID	U	RU
T1	3	0
T3	13	0
T4	9	0
T6	9	0
T7	5	0
SUM	39	0

c, e		
TID	U	RU
T2	12	0
T3	4	0
T4	6	0
T5	5	0
T8	5	0
SUM	32	0

c, f		
TID	U	RU
T3	6	0
T7	4	0
T8	3	0
SUM	13	0

d, e		
TID	U	RU
T3	15	6
T4	9	3
SUM	24	0

d, f		
TID	U	RU
T3	17	9
T7	7	6
SUM	24	0

e, f		
TID	U	RU
T3	8	6
T8	4	7
SUM	12	0

Hình 3. Các utility-list có kích thước k = 2

- **Khởi tạo cấu trúc utility-list với kích thước k = 3 bằng phép nối 3 utility-list với kích thước k = 1, hoặc utility-list với k = 2 nối với của utility-list có kích thước k = 1 có cùng tiền tố là itemset.**

  Ví dụ utility-list({a, b, c}) thì ta lấy utility-list({a}) + nối + utility-list({b}) + nối + utility-list({c}) hoặc utility-list({a, b} + utility-list({c}).

a, b, c		
TID	U	RU
T3	10	0
T7	8	0
T8	11	0
SUM	29	0

a, b, d		
TID	U	RU
T3	21	1
T7	11	1
SUM	32	2

a, b, e		
TID	U	RU
T3	12	1
T8	12	2
SUM	24	3

a, b, f		
TID	U	RU
T3	14	1
T7	10	1
T8	10	2
SUM	34	4

a, c, d		
TID	U	RU
T1	8	0
T3	18	0
T6	24	0
T7	10	0
SUM	60	0

a, c, e		
TID	U	RU
T2	24	0
T3	9	0
T8	10	0
SUM	43	0

a, c, f		
TID	U	RU
T3	11	0
T7	9	0
T8	8	0
SUM	28	0

a, d, e		
TID	U	RU
T3	20	1
SUM	20	1

a, d, f		
TID	U	RU
T3	22	1
T7	12	1
SUM	34	2

b, c, d		
TID	U	RU
T3	15	0
T4	17	0
T7	9	0
SUM	41	0

b, c, e		
TID	U	RU
T3	8	0
T4	15	0
T5	9	0
T7	9	0
SUM	41	0

b, c, f		
TID	U	RU
T3	10	0
T7	6	0
T8	7	0
SUM	23	0

c, d, e		
TID	U	RU
T3	16	0
T4	12	0
SUM	28	0

c, d, f		
TID	U	RU
T3	18	0
T7	8	0
SUM	26	0

d, e, f		
TID	U	RU
T3	20	6
SUM	20	6

Hình 4. Các utility-list có kích thước k = 3

- **Khởi tạo cấu trúc utility-list với kích thước k = 4 bằng phép nối 4 utility-list với kích thước k = 1, hoặc utility-list với k = 3 nối với của utility-list có kích thước k = 1 có cùng tiền tố là itemset.**

a, b, d, e		
TID	U	RU
T3	24	0
SUM	24	0

a, b, d, f		
TID	U	RU
T3	26	1
T7	14	1
SUM	40	2

a, b, c, d		
TID	U	RU
T3	22	0
T7	12	0
SUM	34	0

a, b, c, e		
TID	U	RU
T3	13	0
T8	14	0
SUM	27	0

a, b, c, f		
TID	U	RU
T3	15	0
T7	11	0
T8	12	0
SUM	38	0

Hình 5. Các utility-list có kích thước k = 4

- **Khởi tạo cấu trúc utility-list với kích thước k = 5 bằng phép nối 5 utility-list với kích thước k = 1, hoặc utility-list với k = 4 nối với của utility-list có kích thước k = 1 có cùng tiền tố là itemset.**

a, b, c, d, e		
TID	U	RU
T3	25	0
SUM	25	0

a, b, c, d, f		
TID	U	RU
T3	27	0
T7	15	0
SUM	42	0

a, c, d, e, f		
TID	U	RU
T3	26	0
SUM	26	0

a, b, d, e, f		
TID	U	RU
T3	29	0
SUM	29	0

Hình 6. Các utility-list có kích thước k = 5

- **Khởi tạo cấu trúc utility-list với kích thước k = 6 bằng phép nối 6 utility-list với kích thước k = 1, hoặc utility-list với k = 5 nối với của utility-list có kích thước k = 1 có cùng tiền tố là itemset.**

a, b, c, d, e, f		
TID	U	RU
T3	30	0
SUM	30	0

Hình 7. Các utility-list có kích thước k = 6

- **Kết quả:**
  - {a} = 45 (utility).
  - {a, c} = 59 (utility).
  - {a, d} = 54 (utility).
  - {a, c, d} = 60 (utility).
  - {a, c, e} = 41 (utility).
  - {b, c, d} = 41 (utility).
  - {b, c, e} = 40 (utility).
  - {a, b, d, f} = 40 (utility).
  - {b, c, d, e} = 40 (utility).
  - {a, b, c, d, e, f} = 42 (utility).

CHƯƠNG 3. KHAI THÁC TẬP MỤC HỮU ÍCH CAO BẰNG THUẬT GIẢI DI TRUYỀN

3.1. Vấn đề chung gặp phải của các thuật toán khai thác tập hữu ích cao

Việc khai thác tập mục hữu ích cao bằng cách sử dụng các phương pháp hay thuật toán phổ biến có thể dẫn đến tiêu tốn nhiều về chi phí như bộ nhớ hay thời gian.

Vì vậy cần tìm một giải pháp để có thể cải thiện vấn đề trên bằng cách sử dụng thuật giải di truyền để tìm ra hướng tiếp cận giải quyết vấn đề về thời gian. Mặc dù kết quả khai thác có thể không đầy đủ khi quyết định các tham số để chạy thuật toán không phù hợp, nhưng khi qua nhiều lần thử nghiệm vẫn có thể đưa ra được một lời giải gần đúng.

Vậy nên em quyết định sử dụng các phép toán trên bit, để tối ưu việc tính toán độ hữu ích của các các thể trên cơ sở dữ liệu. Với độ dài nhiễm sắc thể và số mục mặt hàng xuất hiện càng nhiều trên một giao tác trong cơ sở dữ liệu thì áp khi chuyển đổi sang bit để thực hiện phép toán & (AND bitwise) cho nhiễm sắc thể và các giao tác trong cơ sở dữ liệu thì sẽ giúp giảm bớt đi n chiều dài lần số lần so sánh trên các bit 1 của cấu những loại cấu trúc dữ liệu truyền thống như mảng, danh sách, từ điển, …

Tương tự với độ lớn (số hàng) của cơ sở dữ liệu lớn thì khi việc áp dụng lập trình song song để phân mảnh các cơ sở dữ liệu theo chiều ngang, để tính toán độ hữu ích của nhiều NST cùng lúc trên những mảnh khác nhau sau đó hội kết quả lại thay vì tính toán độ hữu ích của 1 NST trên từng giao dịch (hàng) của CSDL.

3.2. Một số hạn chế của thuật toán HUI – Miner và lý do lựa chọn thuật giải di truyền để khai thác tập mục hữu ích cao

Khả năng tìm kiếm tối ưu:

- HUI – Miner: Sử dụng phương pháp duyệt theo tầng, có thể bỏ lỡ một số HUI tiềm năng do bị giới hạn bởi cấu trúc dữ liệu danh sách lợi ích (utility-list).
- Thuật giải di truyền: Sử dụng cơ chế tìm kiếm ngẫu nhiên kết hợp với các toán tử di truyền như lai ghép, đột biến, có khả năng khám phá không gian tìm kiếm rộng hơn và tiềm kiếm ra các HUI tối ưu hơn.

Hiệu quả của việc xử lý dữ liệu lớn:

- HUI – Miner: Khi xử lý tập dữ liệu lớn, hiệu suất của thuật toán có thể giảm do cần duy trì và cập nhật các utility cho tất cả các tập mục ứng viên.
- Thuật giải di truyền: Có khả năng mở rộng tốt hơn khi xử lý tập dữ liệu lớn nhờ vào cơ chế tìm kiếm ngẫu nhiên và khả năng song song hóa.

Độ phức tạp:

- HUI – Miner: Cấu trúc và cách hoạt động của HUI – Miner tương đối đơn giản, dễ dàng triển khai và áp dụng.
- Thuật giải di truyền: Có khả năng tối ưu hóa nhiều mục tiêu đồng thời, phù hợp với các bài toán HUI phức tạp hơn, nơi cần cân nhắc nhiều yếu tố khác nhau như lợi ích, chi phí, sự hỗ trợ, v.v.

Khả năng thích ứng với dữ liệu nhiễu:

- HUI – Miner: Nhạy cảm với dữ liệu nhiễu, có thể ảnh hưởng đến độ chính xác của kết quả khai thác.
- Thuật giải di truyền: Có khả năng thích ứng tốt hơn với dữ liệu nhiều nhờ vào cơ chế tìm kiếm ngẫu nhiên và khả năng loại bỏ các giải pháp không phù hợp.

Vậy nên mặc dù HUI – Miner có ưu điểm về độ đơn giản và hiệu quả khi xử lý dữ liệu vừa phải, thuật giải di truyền tỏ ra vượt trội hơn về khả năng tìm kiếm tối ưu, xử lý dữ liệu lớn, giải quyết bài toán đa mục tiêu và thích ứng với dữ liệu nhiễu.

3.3. Áp dụng tính toán song song để vào thuật giải di truyền

Tính toán song song là một phương pháp sử dụng nhiều tài nguyên tính toán cùng một lúc để giải quyết các bài toán lớn hoặc tốn nhiều thời gian. Thay vì chạy một tác vụ trên một bộ xử lý duy nhất, tính toán song song cho phép phân chia công việc thành nhiều phần nhỏ hơn và thực hiện chúng đồng thời trên nhiều bộ xử lý, máy tính, hoặc cụm máy tính.

Sử dụng nhiều bộ xử lý độc lập để thực hiện các phép tính cùng một lúc. Áp dụng vào bài toán hiện tại. Vấn đề tiêu tốn thời gian nhất là khi tính toán độ thích nghi của các cá thể. Vậy nên ta sẽ chia các number worker của CPU để tính toán hữu ích của cá thể

trên mỗi đoạn phân mảnh của cở sở dữ liệu, sau đó hội kết quả lại để tính ra hữu ích tổng của một cá thể.

Ví dụ:

- Số Number worker của CPU hiện tại là 2
- NST:

STT	Bits				
1	1	0	1	0	1
2	0	1	0	1	1

- Cho cơ sỡ dữ liệu giao tác bên dưới:

TID	Items				
	1	2	3	4	5
T1	0	1	1	1	0
T2	1	1	1	1	1
T3	1	0	1	1	0
T4	0	0	1	1	1
T5	1	1	0	1	1
T6	1	1	1	1	0

- Số tổng số giao dịch là 6 (T1 -> T6) và number worker của CPU là 2 nên ta chia CSDL thành 2 mảnh:
  - Mảnh 1: T1 -> T3
  - Mảnh 2: T4 -> T6
- Áp dụng vào bài toán ta tính song song độ phổ biến của 2 NST cùng lúc:
  - NST 1: Tính trên mảnh 1, sau đó trên mảnh 2
  - NST 2: Tính trên mảnh 2, sau đó trên mảnh 1
  - Rồi sau đó cộng các kết quả của từng NST trên từng mảnh lại.

3.4. Mã giả

Hàm thích nghi (fitness)

Đầu vào (Input):

- Cá thể – chromosome,
- Danh sách giao tác hữu ích - transactions

Đầu ra: Độ hữu ích của cá thể

```
1.  fitness = 0;
2.  for each transaction in transactions
3.      transactionBits = bit of transaction;
4.      transactionValues = values of transaction;
5.      chromosomeBits = bit of chromosome;
6.      mask = transactionBits & chromosomeBits;
7.      if mask = chromosomeBits do
8.          for i = 0 to length(transactionBits) do
9.              if transactionBits[i] = 1 do
10.                 fitness = fitness + transactionValues[i + 1];
11. return fitness;
```

Khởi tạo quần thể ban đầu (Initial Population)

Đầu vào: Kích thước quần thể – populationSize.

Đầu ra: Danh sách các các thể – population.

```
// Khởi tạo danh sách rỗng
1.  population = [];
2.  while length(population) < populationSize do
3.      Khởi tạo các thể với toàn bộ bit bằng 0;
4.      Bật ngẫu nhiên một bit của cá thể lên 1;
5.      Tính độ hữu ích của cá thể;
```

6. Lưu cá thể vào population;
7. **endWhile;**
8. **return** population;

Khai thác tập mục hữu ích cao bằng thuật giải di truyền (GA for HUIM).

Đầu vào (Input):

- Cở sở dữ liệu hữu ích – dataset,
- Số thế hệ – generation,
- Kích thước quần thể – populationSize,
- Tỷ lệ lai ghép – crossoverProbability,
- Tỷ lệ đột biến – mutationProbability,
- Hữu ích tối thiểu – minUtility,

Đầu ra: Các tập mục hữu ích cao – HUI

// Khởi tạo (Initialization)

1. Tạo ngẫu nhiên số lượng populationSize cá thể;
2. Tính toán hữu ích của các cá thể theo hàm thích nghi;
3. Lưu các các thể vào population;

// Lặp theo số lượng thế hệ

4. **for** i = 1 **to** generation **do**

// Quần thể con bằng một nửa quần thể hiện tại

5. subPopulation = ½ population;
6. **while length(**subPopulation**)** < populationSize **do**

// Lựa chọn ngẫu nhiên 2 các thể cha mẹ từ population

7. parent1, parent2 **from** population;

// Lai ghép

```
8.          if số ngẫu nhiên < crossoverProbability do
9.              Tạo ra child1, child2 từ lai gép 1 điểm từ parent1 và parent2;
10.             Tính toán hữu ích của child1, child2;
11.             if child1 && child2 not in subPopulation do
12.                 Lưu child1, child2 vào subPopulation;
13.         else
14.             Lưu parent1, parent2 vào subPopulation;
            // Đột biến
15.         // Lựa chọn ngẫu nhiên một cá thể từ population
16.         chromosomeToMutate from population;
17.         if số ngẫu nhiên < mutationProbability do
18.             Tạo ra mutatedChromosomo từ nghịch đảo một bit ngẫu nhiên trên
                bit của chromosome;
19.             Tính toán hữu ích của mutatedChromosome;
20.             if mutatedChromosome not in subPopulation do
21.                 Lưu mutatedChromosome vào subPopulation;
22.         else
23.             Lưu chromosomeToMutate vào subPopulation;
24.     endWhile;
25.     // Cập nhật subPopulation
26.     subPopulation = subPopulation + population;
27. endFor;
```

3.5. Mô tả các bước khai thác tập mục hữu ích cao bằng thuật toán di truyền.

- Bước 1: Quét cơ sở dữ liệu giao dịch, tìm ra số lượng các mục riêng biệt. Để gán cho chiều dài nhiễm sắc thể (bit) – bitLength.
- Bước 2: Tạo một đối tượng lưu trữ các bit nhiễm sắc thể có kích thước bằng với số lượng các tập mục riêng biệt từ bước 1, và lưu trữ thêm giá trị của các mục của giao dịch đó.

- Bước 3: Tính số lượng trung bình các mục ở toàn bộ giao dịch trong cơ sở dữ liệu – avgLengthItem.
- Bước 4: Khởi tạo ngẫu nhiên các bit nhiễm sắc thể với chiều dài bitLength là 0. Lấy ra ngẫu nhiên số avgLengthItem vị trí từ 1 đến bitLength, sau đó bật các bit của nhiễm sắc thể này lên 1 từ những vị trí ngẫu nhiên trên và lưu các nhiễm sắc thể vào biến quần thể – populations.
- Bước 5: Tính độ hữu ích của các nhiễm sắc thể bằng hàm thích nghi. Nếu độ hữu ích của nhiễm sắc thể lớn hơn hữu ích tối thiểu thì lưu nhiễm sắc thể đó kết quả – huiSet.
- Bước 6: Kiểm tra số lượng nhiễm sắc thể trong population có bằng với populationSize không? Nếu có chuyển qua bước 7, ngược lại thì tiếp tục bước 4.
- Bước 7: Bước vào lần lặp số thế hệ – generations.
- Bước 8: Ở mỗi thế hệ, lấy một nửa các nhiễm sắc thể từ population và lưu vào quần thể con – subPopulation.
- Bước 8: Lựa chọn ngẫu nhiên 2 cá thể từ phần còn lại của population không được lưu vào subPopulation.
- Bước 9: Thực hiện lai ghép trên 2 nhiễm sắc thể trên để tạo ra 2 nhiễm sắc thể con mới.
- Bước 10: Tính độ hữu ích của 2 nhiễm sắc thể con. Nếu độ hữu ích của nhiễm sắc thể nào cao hơn ngưỡng hữu ích tối thiểu thì lưu nhiễm sắc thể đó vào kết quả – huiSet.
- Bước 11: Kiểm tra nhiễm sắc thể đó đã từng đã tồn tại trong subPopulation chưa? Nếu chưa thì lưu nhiễm sắc thể vào mảng subPopulation.
- Bước 12: Chọn ngẫu nhiên 1 nhiễm sắc thể từ phần còn lại của population không được lưu vào subPopulation.
- Bước 13: Tiến hành đột biến nhiễm sắc thể được lựa chọn để tạo ra nhiễm sắc thể mới.
- Bước 14: Tính toán độ hữu ích của nhiễm sắc thể đã đột biến. Nếu độ hữu ích của nhiễm sắc thể đột biến lớn hơn ngưỡng tối thiểu thì lưu nhiễm sắc thể vào – huiSet.

- Bước 15: Kiểm tra nhiễm sắc thể đã đột biến có tồn tại trong subPopulation không? Nếu không thì lưu nhiễm sắc thể đột biến vào subPopulation.
- Bước 16: Kiểm tra kích thước subPopulation đã bằng populationSize chưa? Nếu chưa thì lặp lại bước 8, ngược lại thì qua bước 17.
- Bước 17: Lưu thêm các nhiễm sắc thể từ population vào subPopulation.
- Bước 18: Kiểm tra số lần lặp thế hệ đã bằng generation chưa? Nếu chưa thì lặp lại bước 7, ngược lại thì kết thúc thuật toán và trình bày kết quả từ huiSet.

3.6. Minh họa khai thác tập mục hữu ích cao bằng thuật toán di truyền.

Với các tham số sau:

- Hữu ích tối thiểu (minUtility): 40.
- Số thế hệ (generations): 2.
- Kích thước quần thể (populationSize): 7

Bảng 6. Dữ liệu giao dịch hữu ích từ bảng 1 và bảng 2 (IU * EU)

TID	Hữu ích của item từng giao dịch (U)							TU
	a	b	c	d	e	f	g	
T1	5	0	1	2	0	0	0	8
T2	10	0	6	0	6	0	5	27
T3	5	4	1	12	3	5	0	30
T4	0	8	3	6	3	0	0	20
T5	0	4	2	0	3	0	2	11
T6	15	0	3	6	0	0	0	24
T7	5	2	1	4	0	3	0	15
T8	5	4	2	0	3	1	0	15
SUM	45	22	19	30	18	9	7	150

Diễn giải: Dữ liệu đầu vào gồm các cột sau:

- TID: Mã định danh duy nhất của giao dịch (hoặc số thứ tự của mỗi giao dịch).
- TU: Độ hữu ích của cả giao dịch.

- U: Hữu ích của mỗi item trong giao dịch được tính bằng công thức sau $U(\{item\}, T_n) = IU(\{item\}, T_n) * EU(\{item\})$

Bảng 7. Biến đổi dữ liệu giao dịch hữu ích từ bảng 9 thành dữ liệu dạng bit

TID	danh sách bit theo item của giao dịch						
	a	b	c	d	e	f	g
T1	1	0	1	1	0	0	0
T2	1	0	1	0	1	0	1
T3	1	1	1	1	1	1	0
T4	0	1	1	1	1	0	0
T5	0	1	1	0	1	0	1
T6	1	0	1	1	0	0	0
T7	1	1	1	1	0	1	0
T8	1	1	1	0	1	1	0

Bài giải:

1. Khởi tạo quần thể ban đầu với kích thước quần thể là 7 và lưu vào population.

Bảng 8. Danh sách các ứng viên với kích thước quần thể là 7

STT	Các ứng viên						
	a	b	c	d	e	f	g
C1	1	0	0	0	0	0	0
C2	1	1	0	0	0	0	0
C3	1	0	0	0	0	1	0
C4	0	1	0	0	0	0	0
C5	1	0	1	0	0	0	0
C6	1	0	1	1	0	0	0
C7	0	1	1	0	0	0	0

2. Tính độ hữu ích của các ứng viên trong quần thể qua hàm thích nghi từ công thức
   - $U(X) = \sum_{T \in DB \wedge X \subseteq T} u(X, T)$
   - Và toán tử AND bitwise (&).
   - Nếu ta có mask = bit của ứng viên (C) & bit của giao dịch (T). Nếu mask bằng với bit của ứng viên thì ta tổng các hữu ích của item trong ứng viên ứng với hữu ích của các giao tác bằng với mask trong CSDL.

Xét C1

Bảng 9. Bit ứng viên C1 & bit của giao dịch T trong CSDL

Xét	Bit của ứng viên (C1)								Bit của giao tác (T)						
C1 & T1	1	0	0	0	0	0	0	&	1	0	1	1	0	0	0
C1 & T2	1	0	0	0	0	0	0		1	0	1	0	1	0	1
C1 & T3	1	0	0	0	0	0	0		1	1	1	1	1	1	0
C1 & T4	1	0	0	0	0	0	0		0	1	1	1	1	0	0
C1 & T5	1	0	0	0	0	0	0		0	1	1	0	1	0	1
C1 & T6	1	0	0	0	0	0	0		1	0	1	1	0	0	0
C1 & T7	1	0	0	0	0	0	0		1	1	1	1	0	1	0
C1 & T8	1	0	0	0	0	0	0		1	1	1	0	1	1	0

Bảng 10. Mask của C1 & T và hữu ích của ứng viên C1

Xét	Mask (C1 & T)								Hữu ích của ứng viên (C1) trong CSDL						
U(C1, T1)	1	0	0	0	0	0	0		5	0	0	0	0	0	0
U(C1, T2)	1	0	0	0	0	0	0		10	0	0	0	0	0	0
U(C1, T3)	1	0	0	0	0	0	0		5	0	0	0	0	0	0
U(C1, T4)	0	0	0	0	0	0	0		0	0	0	0	0	0	0
U(C1, T5)	0	0	0	0	0	0	0	=	0	0	0	0	0	0	0
U(C1, T6)	1	0	0	0	0	0	0		15	0	0	0	0	0	0
U(C1, T7)	1	0	0	0	0	0	0		5	0	0	0	0	0	0
U(C1, T8)	1	0	0	0	0	0	0		5	0	0	0	0	0	0
SUM									45						

Xét C2

Bảng 11. Bit ứng viên C2 & bit của giao dịch T trong CSDL

Xét	Bit của ứng viên (C2)						
C2 & T1	1	1	0	0	0	0	0
C2 & T2	1	1	0	0	0	0	0
C2 & T3	1	1	0	0	0	0	0
C2 & T4	1	1	0	0	0	0	0
C2 & T5	1	1	0	0	0	0	0
C2 & T6	1	1	0	0	0	0	0
C2 & T7	1	1	0	0	0	0	0
C2 & T8	1	1	0	0	0	0	0

&

Bit của giao tác (T)						
1	0	1	1	0	0	0
1	0	1	0	1	0	1
1	1	1	1	1	1	0
0	1	1	1	1	0	0
0	1	1	0	1	0	1
1	0	1	1	0	0	0
1	1	1	1	0	1	0
1	1	1	0	1	1	0

Bảng 12. Mask của C2 & T và hữu ích của ứng viên C2

Xét	Mask (C2 & T)								Hữu ích của ứng viên (C2) trong CSDL						
U(C2, T1)	1	0	0	0	0	0	0	=	0	0	0	0	0	0	0
U(C2, T2)	1	0	0	0	0	0	0		0	0	0	0	0	0	0
U(C2, T3)	1	1	0	0	0	0	0		5	4	0	0	0	0	0
U(C2, T4)	0	1	0	0	0	0	0		0	0	0	0	0	0	0
U(C2, T5)	0	1	0	0	0	0	0		0	0	0	0	0	0	0
U(C2, T6)	1	0	0	0	0	0	0		0	0	0	0	0	0	0
U(C2, T7)	1	1	0	0	0	0	0		5	2	0	0	0	0	0
U(C2, T8)	1	1	0	0	0	0	0		5	4	0	0	0	0	0
SUM									25						

Xét C3

Bảng 13. Bit ứng viên C3 & bit của giao dịch T trong CSDL

Xét	Bit của ứng viên (C3)								Bit của giao tác (T)						
C3 & T1	1	0	0	0	0	1	0	&	1	0	1	1	0	0	0
C3 & T2	1	0	0	0	0	1	0		1	0	1	0	1	0	1
C3 & T3	1	0	0	0	0	1	0		1	1	1	1	1	1	0
C3 & T4	1	0	0	0	0	1	0		0	1	1	1	1	0	0
C3 & T5	1	0	0	0	0	1	0		0	1	1	0	1	0	1
C3 & T6	1	0	0	0	0	1	0		1	0	1	1	0	0	0
C3 & T7	1	0	0	0	0	1	0		1	1	1	1	0	1	0
C3 & T8	1	0	0	0	0	1	0		1	1	1	0	1	1	0

Bảng 14. Mask của C3 & T và hữu ích của ứng viên C3

Xét	Mask (C3 & T)						
U(C3, T1)	1	0	0	0	0	0	0
U(C3, T2)	1	0	0	0	0	0	0
U(C3, T3)	1	0	0	0	0	1	0
U(C3, T4)	0	0	0	0	0	0	0
U(C3, T5)	0	0	0	0	0	0	0
U(C3, T6)	1	0	0	0	0	0	0
U(C3, T7)	1	0	0	0	0	1	0
U(C3, T8)	1	0	0	0	0	1	0
SUM							

=

Hữu ích của ứng viên (C3) trong CSDL						
0	0	0	0	0	0	0
0	0	0	0	0	0	0
5	0	0	0	0	5	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
5	0	0	0	0	3	0
5	0	0	0	0	1	0
24						

Xét tương tự cho C4 đến C7, ta được các bảng sau:

Bảng 15. Mask của C4, C5, C6, C7 & T và hữu ích của ứng viên C4, C5, C6, C7

Xét	Mask (C4 & T)						
U(C4, T3)	0	1	0	0	0	0	0
U(C4, T4)	0	1	0	0	0	0	0
U(C4, T5)	0	1	0	0	0	0	0
U(C4, T7)	0	1	0	0	0	0	0
U(C4, T8)	0	1	0	0	0	0	0
SUM							

=

Hữu ích của ứng viên (4) trong CSDL						
0	4	0	0	0	0	0
0	8	0	0	0	0	0
0	4	0	0	0	0	0
0	2	0	0	0	0	0
0	4	0	0	0	0	0
22						

Xét	Mask (C5 & T)						
U(C5, T1)	1	0	1	0	0	0	0
U(C5, T2)	1	0	1	0	0	0	0
U(C5, T3)	1	0	1	0	0	0	0
U(C5, T6)	1	0	1	0	0	0	0
U(C5, T7)	1	0	1	0	0	0	0
U(C5, T8)	1	0	1	0	0	0	0
SUM							

=

Hữu ích của ứng viên (C5) trong CSDL						
5	0	1	0	0	0	0
10	0	6	0	0	0	0
5	0	1	0	0	0	0
15	0	3	0	0	0	0
5	0	1	0	0	0	0
5	0	2	0	0	0	0
59						

Xét	Mask (C6 & T)						
U(C6, T1)	1	0	1	1	0	0	0
U(C6, T3)	1	0	1	1	0	0	0
U(C6, T6)	1	0	1	1	0	0	0
U(C6, T7)	1	0	1	1	0	0	0
SUM							

=

Hữu ích của ứng viên (C6) trong CSDL						
5	0	1	2	0	0	0
5	0	1	12	0	0	0
15	0	3	6	0	0	0
5	0	1	4	0	0	0
60						

Xét	Mask (C7 & T)						
U(C7, T3)	0	1	1	0	0	0	0
U(C7, T4)	0	1	1	0	0	0	0
U(C7, T5)	0	1	1	0	0	0	0
U(C7, T7)	0	1	1	0	0	0	0
U(C7, T8)	0	1	1	0	0	0	0
SUM							

=

Hữu ích của ứng viên (C7) trong CSDL						
0	4	1	0	0	0	0
0	8	3	0	0	0	0
0	4	2	0	0	0	0
0	2	1	0	0	0	0
0	4	2	0	0	0	0
31						

Thêm các ứng viên vào quần thể của thế hện hiện tại và sắp xếp lại các ứng viên theo thứ tự hữu ích giảm dần.

Bảng 16. Sắp xếp lại thứ tự các ứng viên theo fitness giảm đần

STT	Các ứng viên							Fitness
	a	b	c	d	e	f	g	
C6	1	0	1	1	0	0	0	60
C5	1	0	1	0	0	0	0	59
C1	1	0	0	0	0	0	0	45
C7	0	1	1	0	0	0	0	31
C2	1	1	0	0	0	0	0	25
C3	1	0	0	0	0	1	0	24
C4	0	1	0	0	0	0	0	22

3. Giữ lại các ứng viên mạnh với fitness cao để lai ghép và đột biến.
   - Giữ lại C6, C5, C1, C7 và loại bỏ C3, C4.

Bảng 17. Những ứng viên còn lại qua quá trình chọn lọc

STT	Các ứng viên							Fitness
	a	b	c	d	e	f	g	
C6	1	0	1	1	0	0	0	60
C5	1	0	1	0	0	0	0	59
C1	1	0	0	0	0	0	0	45
C7	0	1	1	0	0	0	0	31
C2	1	1	0	0	0	0	0	25

4. Bắt đầu sản sinh ứng viên qua từng thế hệ (Lần đầu tiên generation = 1).
5. Lấy ra ngẫu nhiên 2 ứng viên: C1 và C7.
   - Chọn ngẫu nhiên 1 điểm trong NST của cá thể: Vị trí thứ 2.
   - Chia đoạn nhiễm sắc thể cha mẹ thành 2 đoạn tương ứng với vị trí thứ 2.
   - Sau đó nối đoạn 1 của NST C1 với đoạn 2 của NST 2, và ngược lại để tạo ra 2 NST con cái mới.

Bảng 18. Quá trình lai ghép 2 NST

C1	1	0	0	0	0	0	0
C7	0	1	1	0	0	0	0

Đoạn 1 - C1	1	0			
Doạn 2 - C1	0	0	0	0	0

Đoạn 1 - C7	0	1			
Doạn 2 - C7	1	0	0	0	0

Child 1 (C8)	1	0	1	0	0	0	0
Child 2	0	1	0	0	0	0	0

6. Chọn NST con thứ 2 để tiến hành đột biến:
   - Chọn ví trí để đột biến: Vị trí 4.

Bảng 19. Quá trình đột biến NST C9 để tạo ra ứng viên mới C10

Child 2	0	1	0	0	0	0	0
Child 2 đã đột biến (C9)	0	1	0	1	0	0	0

- Tính độ hữu ích của các NST mới bằng hàm mục tiêu:
  - Xét C8, C9 và tính độ hữu ích của 2 ứng viên

Bảng 20. Mask của C8, C9 & T và hữu ích của ứng viên C8, C9

Xét	Mask (C8 & T)								Hữu ích của ứng viên (C8) trong CSDL						
U(C8, T1)	1	0	1	0	0	0	0	=	5	0	1	0	0	0	0
U(C8, T2)	1	0	1	0	0	0	0		10	0	6	0	0	0	0
U(C8, T3)	1	0	1	0	0	0	0		5	0	1	0	0	0	0
U(C8, T6)	1	0	1	0	0	0	0		15	0	3	0	0	0	0
U(C8, T7)	1	0	1	0	0	0	0		5	0	1	0	0	0	0
U(C8, T8)	1	0	1	0	0	0	0		5	0	2	0	0	0	0
SUM									59						
Xét	Mask (C9 & T)								Hữu ích của ứng viên (C9) trong CSDL						
U(C9 T3)	0	1	0	1	0	0	0	=	0	4	0	12	0	0	0
U(C9, T4)	0	1	0	1	0	0	0		0	8	0	6	0	0	0
U(C9, T7)	0	1	0	1	0	0	0		0	2	0	4	0	0	0
SUM									36						

- Thêm 2 ứng viên mới vào quần thể và sắp xếp lại ứng viên qua quá trình đột biến và lai ghép

Bảng 21. Quần thể hiện tại sau khi thêm 2 ứng viên mới

STT	Các ứng viên							Fitness
	a	b	c	d	e	f	g	
C6	1	0	1	1	0	0	0	60
C5	1	0	1	0	0	0	0	59
C8	1	0	1	0	0	0	0	59
C1	1	0	0	0	0	0	0	45
C9	0	1	0	1	0	0	0	36
C7	0	1	1	0	0	0	0	31
C2	1	1	0	0	0	0	0	25

7. Kiểm tra tổng số ứng viên trong quần thể đã bằng với populationSize chưa? Nếu chưa thì quay lại bước 5, ngược lại thì qua thế hệ tiếp theo (generation hiên tại + 1).
8. Kiểm tra điều kiện dừng nếu thế hệ hiện tại (generation) lớn hơn hoặc bằng generations thì dừng lại và qua bước 9, nếu nhỏ hơn thì ta quay lại bước 4.
9. Trình bày kết quả: Các cá thể có fitness >= 40 (ngưỡng tối thiểu) theo bảng bên dưới

Kết quả:

Bảng 22. Kết quả của khai thác tập mục hữu ích cao bằng thuật giải di truyền

Bit của ứng viên							Dạng itemset	Độ hữu ích
A	B	C	D	E	F	G		
1	0	1	1	0	0	0	{a, c, d}	60
1	0	1	0	0	0	0	{a, c}	59
1	0	0	1	0	0	0	{a, d}	54
1	0	0	0	0	0	0	{a}	45
1	1	1	1	1	1	0	{a, b, c, d, e}	42
0	1	1	1	0	0	0	{b, c, d}	41
1	0	1	0	1	0	0	{a, c, e}	41
0	1	1	0	1	0	0	{b, c, d}	40
1	1	0	1	0	1	0	{a, b, d, f}	40
0	1	1	1	1	1	0	{b, c, d, e}	40

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

4.1. Thông số các loại dữ liệu thử nghiệm

Bảng 23. Bảng kê khai các loại dữ liệu thử nghiệm

Tên tập dữ liệu	Số giao dịch	Số vật phẩm	TB vật phẩm / giao tác	Mật độ (Tỷ trọng)
chainstore	1 112 949	46 086	7.23	Thưa (0.02 %)
Crimes in Chicago	2 662 309	35	1 795	Dày (5.13 %)
mushroom	8 416	119	23	Dày (19.33 %)
connect	67 557	129	43	Dày (33.33 %)
kosarak	990 002	41 270	8.1	Thưa (0.02 %)

4.2. Thông số các loại máy thử nghiệm

Bảng 24. Thông số kỹ thuật những máy thử nghiệm

TSKT	Máy 1	Máy 2	Máy 3
Tên thiết bị	PM12 – PC04	DESKTOP-KKGPLKT	DELL PRECISION – 3520
CPU	12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz	Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.70 GHz	Intel(R) Core(TM) i7-6820 HQ CPU @ 2.70GHz x 8
RAM	32 GB	16.0 GB	16.0 GB
OS	Windows 10 Home Single Language	Windows 10 Pro	Ubuntu 22.04

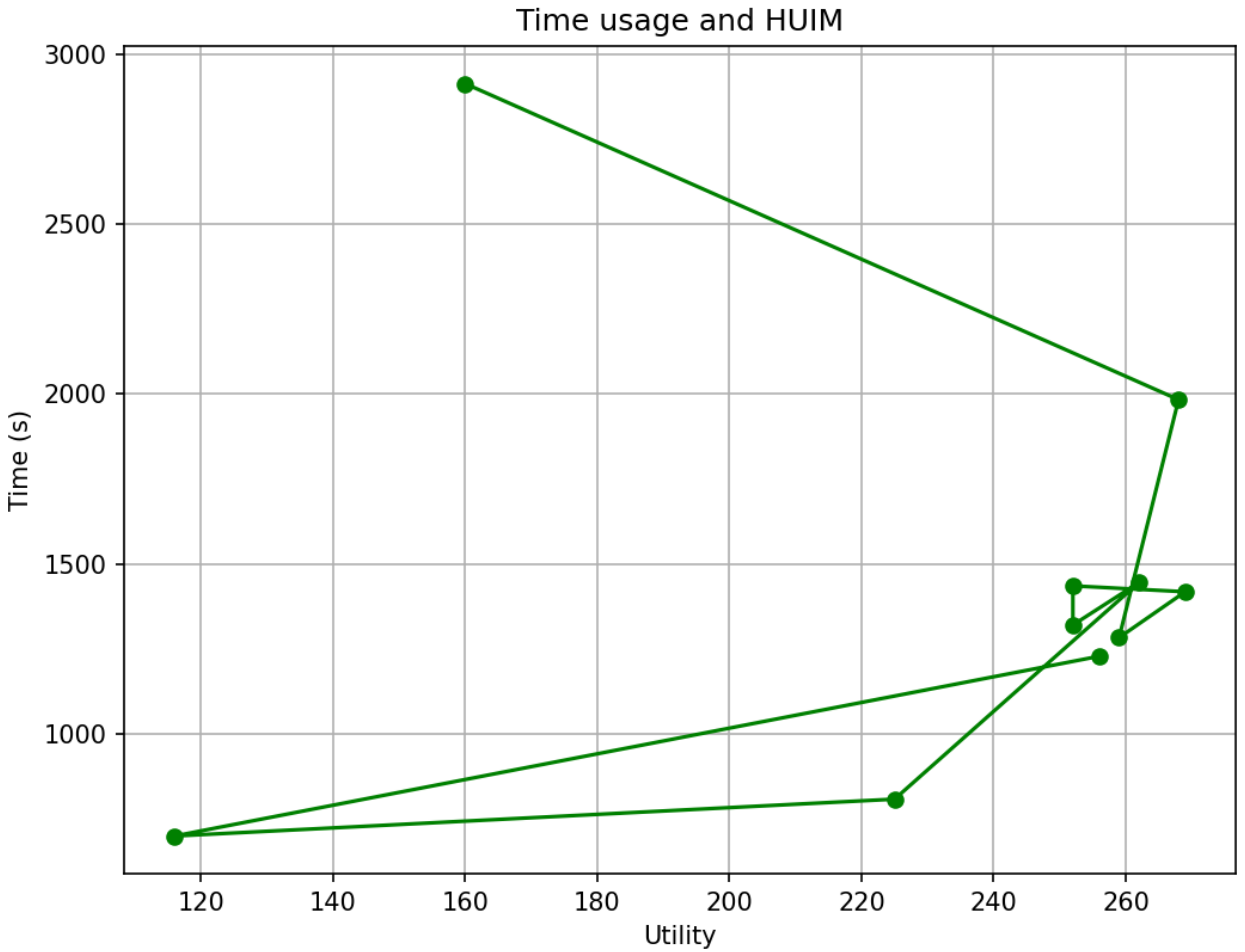
4.3. Kết quả chạy thử nghiệm

4.3.1. Kết quả thử nghiệm trên các máy

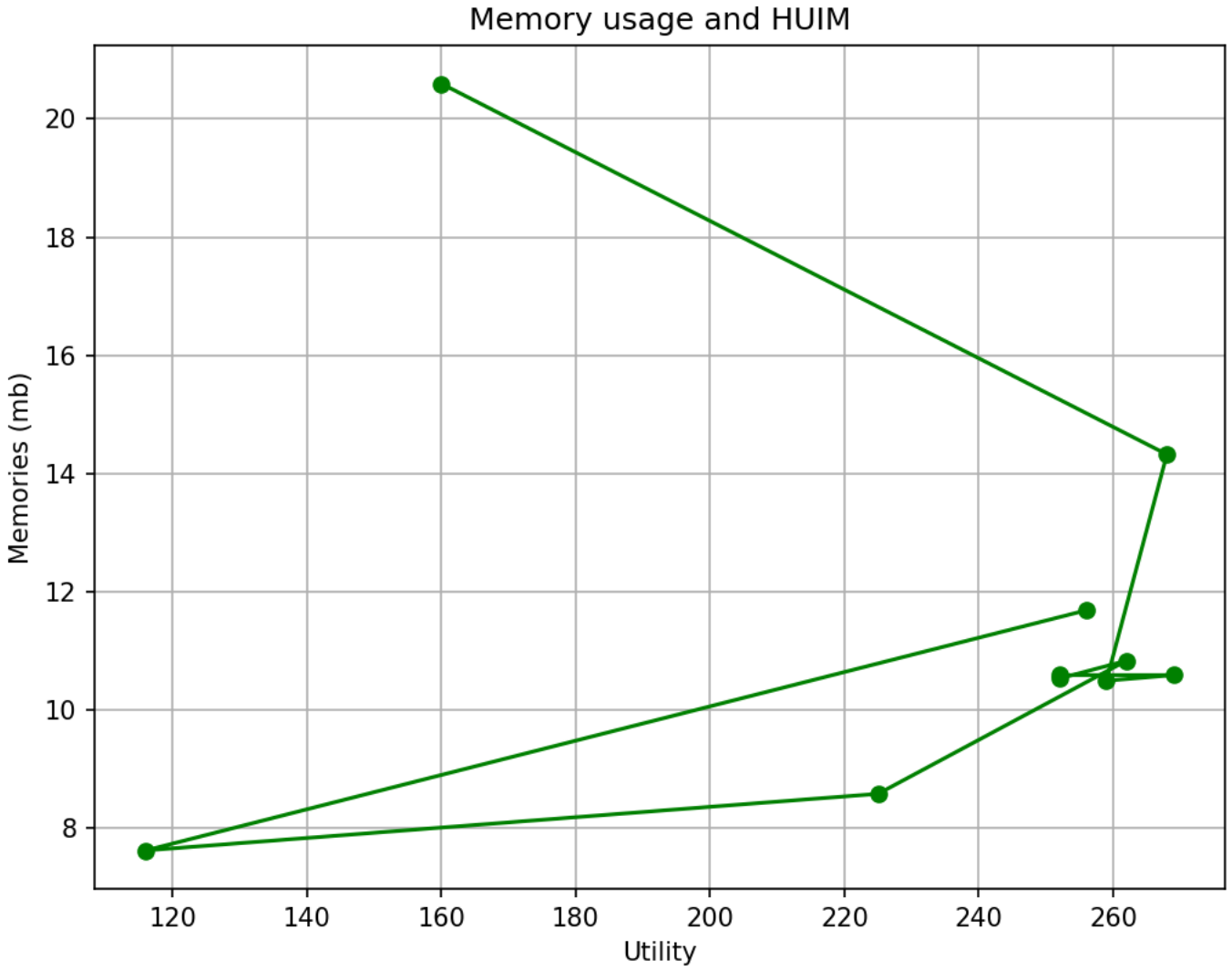
- Dữ liệu chainstore trên Máy 1

Bảng 25. Kết quả tập khai thác từ dữ liệu chainstore

Parameters	TH 1	TH 2	TH 3	TH 4	TH 5	TH 6	TH 7	TH 8	TH 9	TH 10
Thế hệ	20	20	10	20	20	20	20	20	20	20
Kích thước quần thể	200	100	200	200	200	200	200	200	250	500
Tỷ lệ lai ghép	0.1	0.1	0.1	0.1	0.3	0.1	0.05	0.2	0.2	0.1
Tỷ lệ đột biến	0.8	0.8	0.8	0.8	0.8	0.7	0.85	0.7	0.9	0.8
Hữu ích tối thiểu	200	200	200	100	200	200	200	200	300	1000
Tập mục hữu ích	256	116	225	262	252	252	269	259	268	160
Độ dài lớn nhất của tập mục	3	2	2	2	3	3	3	2	3	4
Thời gian (giây)	1228.292	699.349	807.769	1445.172	1320.137	1435.331	1417.951	1283.097	1983.097	2912.923
Bộ nhớ (mb)	11.688	7.625	8.582	10.836	10.532	10.591	10.591	10.496	14.325	20.597
Tham số thay đổi		PS	GEN	MU	MP	CP	MP, CP	MP, CP	All	PS, MU



Hình 8. Biểu đồ chi phí thời gian và số HUI khai thác được từ dữ liệu chainstore

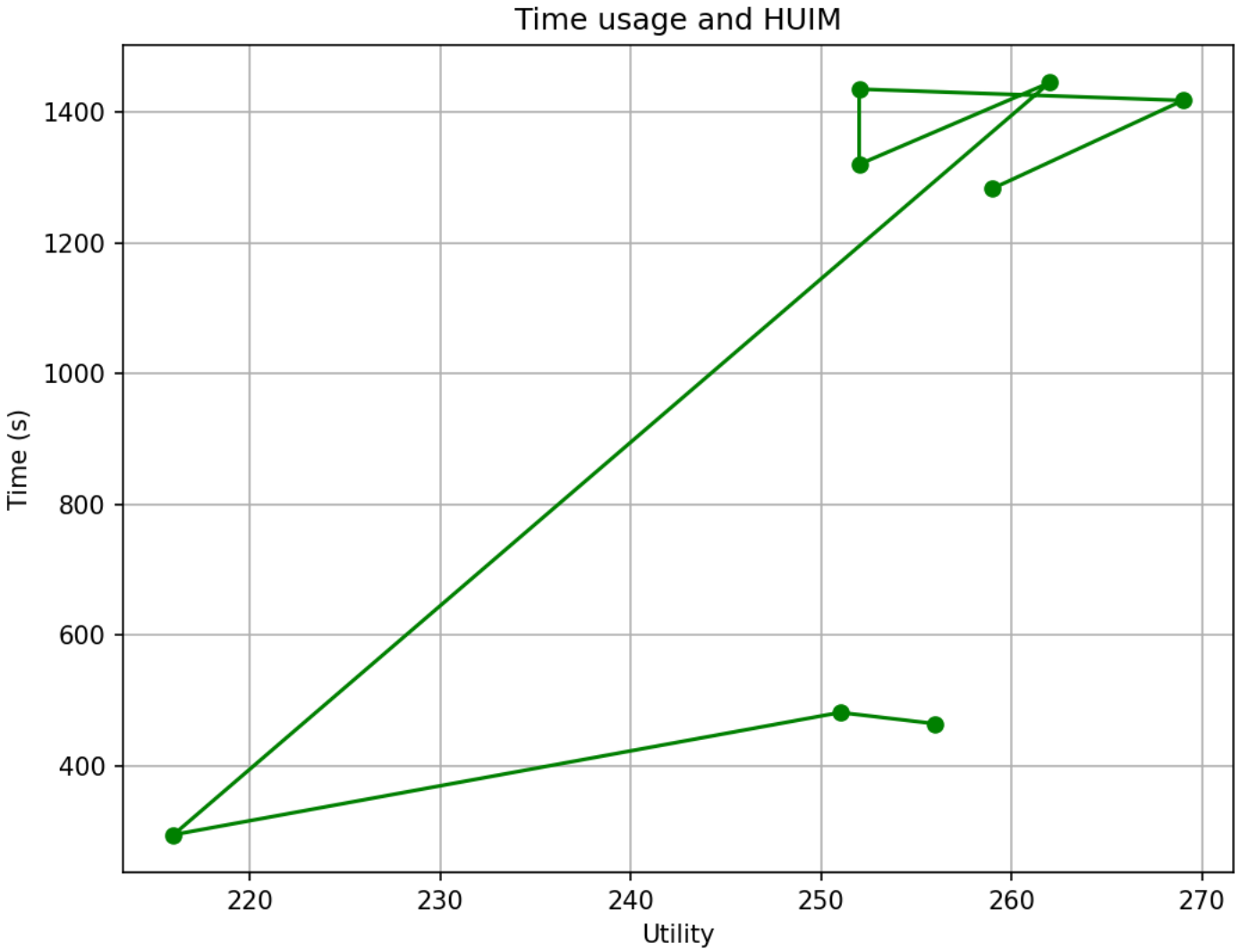


Hình 9. Biểu đồ chi phí bộ nhớ và số HUI khai thác được từ dữ liệu chainstore

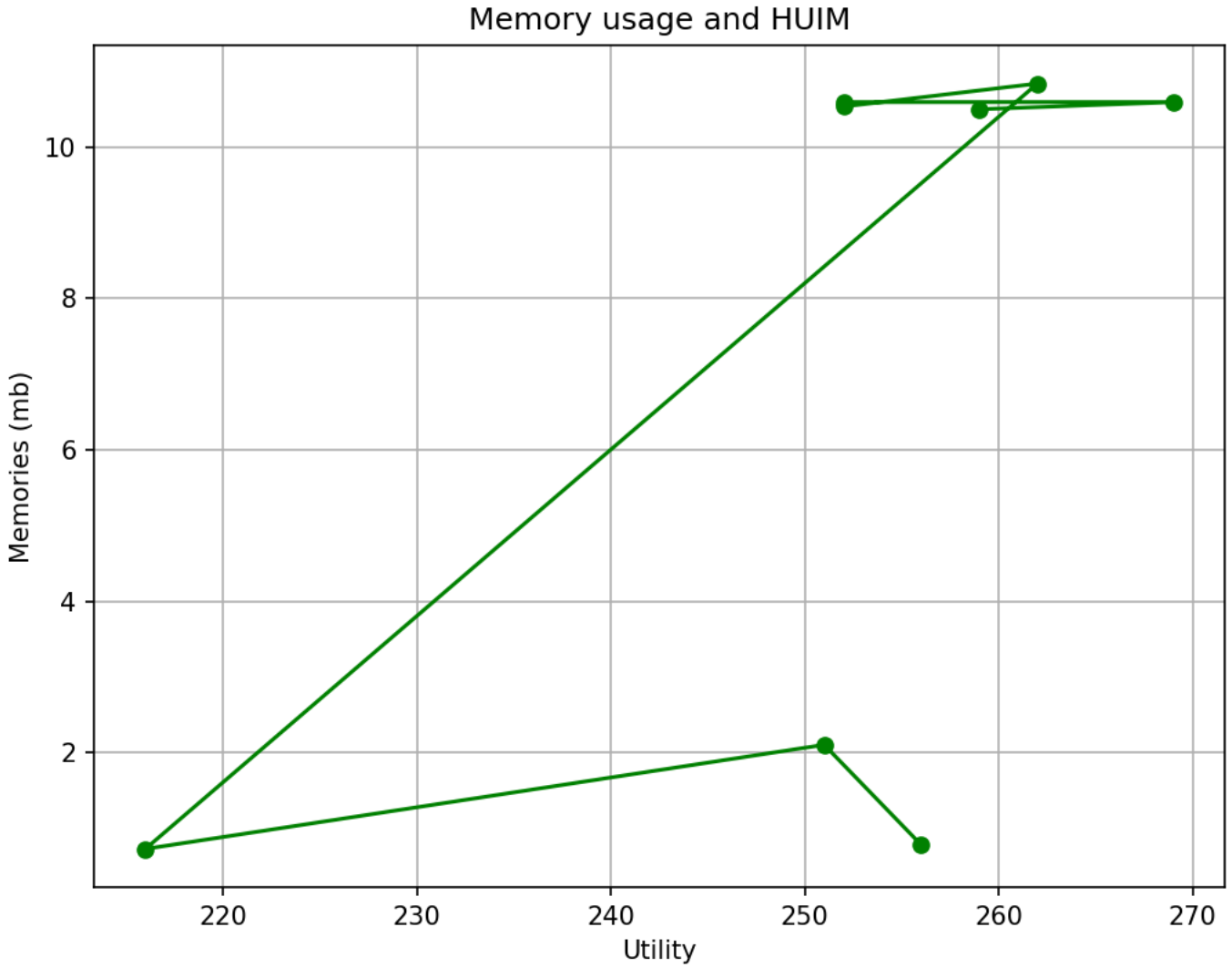
- Dữ liệu thử nghiệm Crimes in Chicago trên máy 1

Bảng 26. Kết quả tập khai thác từ dữ liệu Crimes in Chicago

Tham số	TH 1	TH 2	TH 3	TH 4	TH 5	TH 6	TH 7	TH 8
Thế hệ	500	500	300	20	20	20	20	20
Kích thước quần thể	30	30	30	200	200	200	200	200
Tỷ lệ lai ghép	0.1	0.1	0.1	0.1	0.3	0.1	0.05	0.2
Tỷ lệ đột biến	0.8	0.8	0.8	0.8	0.8	0.7	0.85	0.7
Hữu ích tối thiểu	200	500	200	100	200	200	200	200
Tập mục hữu ích	256	251	216	262	252	252	269	259
Độ dài lớn nhất của tập mục	5	5	5	4	4	5	5	5
Thời gian (giây)	463.987	481.219	294.148	1445.172	1320.137	1435.331	1417.951	1283.097
Bộ nhớ (mb)	0.785	2.105	0.73	10.836	10.532	10.591	10.591	10.496
Tham số thay đổi	PS	MU	GEN	MU	MP	CP	MP, CP	MP, CP



Hình 10. Biểu đồ chi phí thời gian và số HUI khai thác được từ dữ liệu Crimes in Chicago

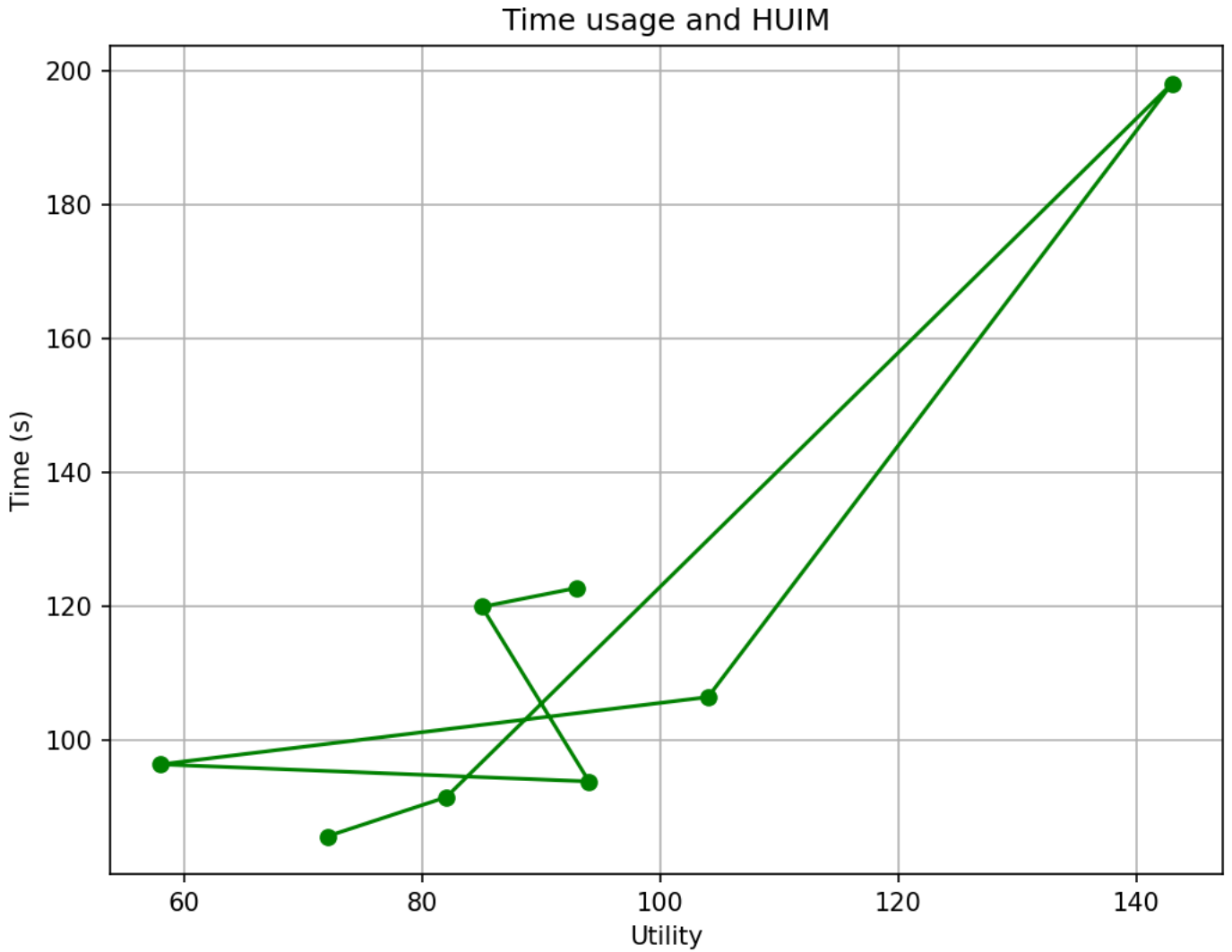


Hình 11. Biểu đồ chi phí bộ nhớ và số HUI khai thác được từ dữ liệu Crimes in Chicago

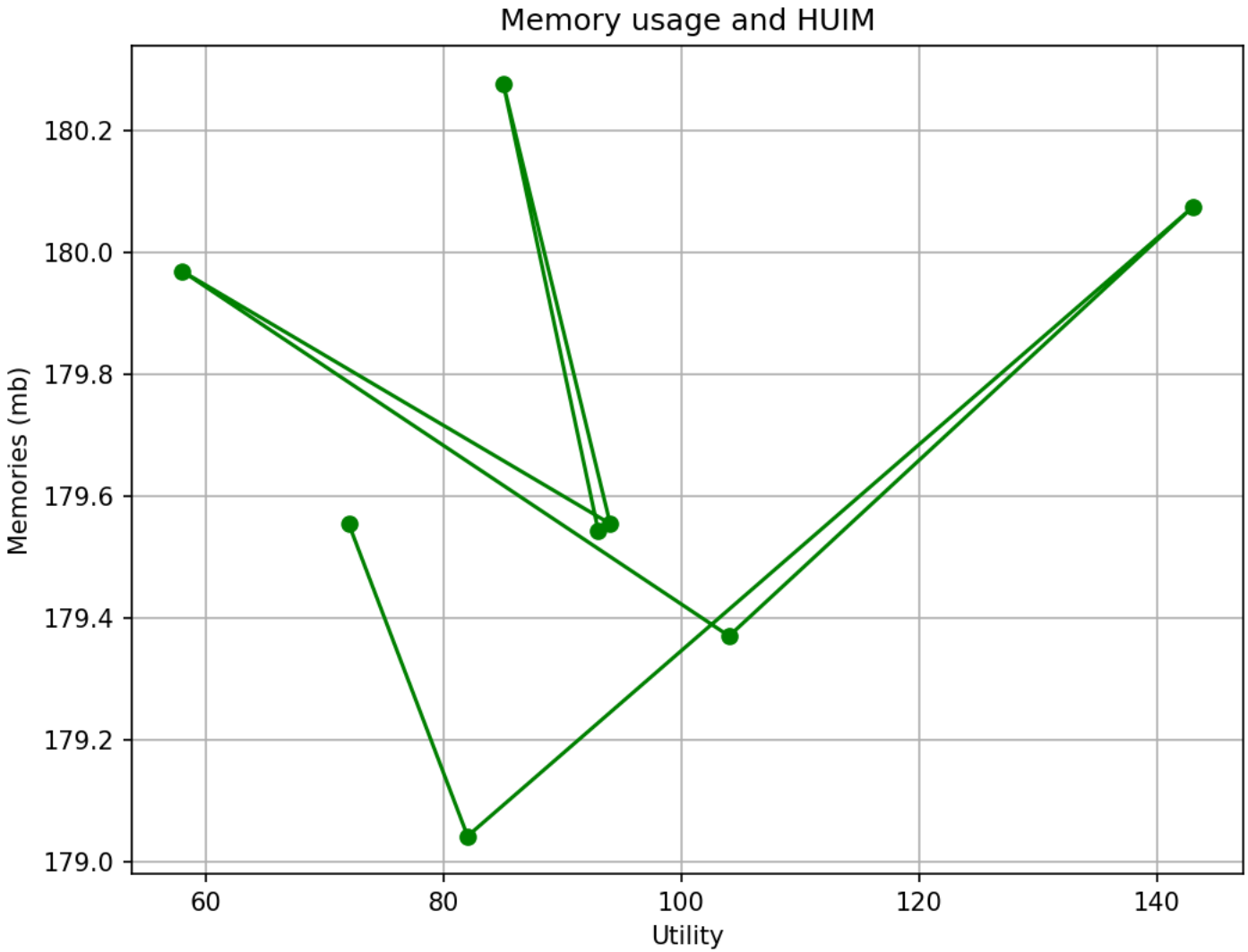
- Dữ liệu thử nghiệm connect trên Máy 1 và Máy 2

Bảng 27. Kết quả tập khai thác từ dữ liệu connect

Tham số	TH 1 – M1	TH 2 – M2	TH 3 – M1	TH 4 – M1	TH 5 – M2	TH 6 – M2	TH 7 – M1	TH 8 – M1
Thế hệ	50	50	100	50	50	50	50	40
Kích thước quần thể	400	300	300	300	300	300	300	400
Tỷ lệ lai ghép	0.1	0.1	0.1	0.3	0.1	0.05	0.2	0.15
Tỷ lệ đột biến	0.8	0.8	0.8	0.8	0.9	0.75	0.9	0.85
Hữu ích tối thiểu	400	300	400	400	400	400	400	500
Tập mục hữu ích	72	82	143	104	58	94	85	93
Độ dài lớn nhất của tập mục	10	12	12	9	8	10	12	13
Thời gian (giây)	85.646	91.498	198.085	106.449	96.396	93.848	119.918	122.752
Bộ nhớ (mb)	179.555	179.043	180.074	179.371	179.969	179.555	180.277	179.543
Tham số thay đổi	PS	MU	GEN	MP	CP	MP, CP	MP, CP	All



Hình 12. Biểu đồ chi phí thời gian và số HUI khai thác được từ dữ liệu connect



Hình 13. Biểu đồ chi phí thời gian và số HUI khai thác được từ dữ liệu connect

4.3.2. Kết luận từ các thử nghiệm

Thời gian, bộ nhớ, độ dài của tập mục hữu ích cao bị ảnh hưởng nhiều bởi 2 tham số thế hệ và kích thước quần thể. Với kích thước cá quần thể lớn và thế hệ sẽ cho ra kết quả nhiều tập mục, và sẽ xuất hiên các tập mục có độ dài lớn hơn, nhưng bên cạnh đó sẽ làm tiêu tốn nhiều thời gian và bộ nhớ của máy.

Thay đổi tham số ngưỡng tối thiểu cao sẽ làm giảm đi số lượng kết quả của tập mục, nhưng bù lại có thể xuất hiện được những tập mục có nhiều mục hữu ích và các mục có khả năng đều phổ biến.

Thay đổi tỷ lệ đột biến và lai ghép nhìn chung không làm ảnh hưởng nhiều đến kết quả về số lượng tập mục xuất hiện, nhưng đối với dữ liệu dày có thể xuất hiện các tập mục với kích thước lớn. Đối với loại dữ liệu thưa thì xuất hiện ít số lượng các tập mục hơn và kích thước của tập mục thường nhỏ. Nhưng nhìn chung thì không có ảnh hưởng nhiều đến chi phí khai thác của thuật toán.

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Qua thời gian tìm hiệu và thực hiện, đề tài nghiên cứu nhìn chung đã đạt được hầu hết các mục tiêu đặt ra, cụ thể như sau:

- Tổng quan về các khái niệm: Khai khoáng dữ liệu, tập mục phổ biến, luật kết hợp, tập mục hữu ích cao.
- Trình bày được hạn chế của việc khai thác tập mục phổ biến và ưu điểm của tập mục hữu ích cao.
- Giới thiệu và minh họa các thuật toán về khai thác tập.
- Tìm hiểu và nghiên cứu các khái niệm, công thức, ưu điểm, hạn chế, … của việc khai thác tập hữu ích cao từ đó đưa ra hướng giải quyết những hạn chế của việc khai thác dữ liệu này.
- Trình bày và minh họa các khái niệm về thuật giải di truyền.
- Áp dụng thuật giải di truyền vào việc khai thác tập mục hữu ích cao.
- Áp dụng các loại cấu trúc dữ liệu như bitarray, numpy, dictionary để cải thiện khả năng xử lý, tối ưu về thời gian và bộ nhớ.
- Tích hợp lập trình song song, xử lý đa luồng để cải thiện vấn đề về tốc độ xử lý tập dữ liệu lớn trong việc khai thác tập mục hữu ích cao.
- Thiết kế giao diện cho ứng dụng để trực quan hơn trong việc hiển thị kết quả.

5.2. Hạn chế

Vì các lý do về thời gian, sự thiếu sót về kiến thức và tài nguyên nên bài nghiên cứu còn tồn đọng một số hạn chế sau:

- Các phần trình bày về lý thuyết vẫn còn đơn giản, mang tính chất giới thiệu chủ yếu.
- Loại dự liệu đầu khai thác vẫn là dữ liệu tĩnh và đã qua xử lý để làm sạch.
- Đối với dữ liệu quá lớn sẽ gây tốn kém về chi phí như thời gian, bộ nhớ.
- Kết quả trả về các tập mục hữu ích cao của việc khai thác tập đối với loại dữ liệu lớn và thưa có thể sẽ không đáp ứng được độ dài mong muốn về kích thước tập.

- Chưa có giải pháp tính toán ra các tham số cụ thể nên vẫn còn phụ thuộc vào số lần thử nghiệm.

5.3. Hướng phát triển

Để phát triển bài nghiên cứu của đề tài hơn, cần xem xét hướng giải quyết các hạn chế trên:

- Tìm hiểu chuyên sâu, và mở rộng hơn về các lý thuyết trên.
- Phát triển thuật toán có thể xử lý được loại dữ liệu động.
- Sử dụng thêm các cấu trúc dữ liệu hiểu quả hơn để tránh việc làm tiêu tốn tài nguyên về bộ nhớ và tốc độ xử lý của thuật toán.
- Phát triển thêm khả năng đọc và xử lý dữ liệu thô để tránh phụ thuộc vào dữ liệu mẫu có sẵn.
- Tìm hiểu các công thức tính toán các tham số của thuật toán như: Hữu ích tối thiểu, tỷ lệ đột biến, tỷ lệ lai ghép, kích thước quần thể và thế hệ.

TÀI LIỆU THAM KHẢO

[1] H. T. Vỹ, L. Q. H. and T. N. Châu, "FHURIM: Thuật toán khai phá tập mục hữu ích cao hiếm," Đà Nẵng; Huế, 2019.

[2] P. D. Thanh and L. T. M. Nguyen, "Thuật toán khai thác TOP-K tập hữu ích cao dựa trên di truyền với đột biến xếp hạng," HUFLIT Journal of Science, Hồ Chí Minh, 2022.

[3] P. Fournier-Viger, C.-W. Wu and S. Z. S., "FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning," ISMIS, 2014.

[4] P. D. Thanh and L. T. M. Nguyen, "Khai thác tập phổ biến từ dữ liệu luồng bằng cách sử dụng thuật toán di truyền," Hồ Chí Minh.

[5] P. D. Thanh, "Luận Văn Thạc Sĩ Phạm Đức Thành," Hồ Chí Minh, 2006.

[6] N. Đ. Sơn and K. K. Trương, "Giải thuật di truyền," Học Viện Công Nghệ Bưu Chính Viễn Thông thành phố Hồ Chí Minh, Hồ Chí Minh, 2022.

[7] J. Carr, "An Introduction to Genetic Algorithms," 16 May 2014. [Online]. Available: https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf.

[8] P. D. Thanh, "Cải tiến thuật toán HMINER cho việc khai thác thác tập hữu ích cao trên tập dữ liệu thưa," Hồ Chí Minh, 2023.

[9] W. Song and C. Huang, "Mining High Utility Itemsets Using Bio-Inspired," IEEE Access, 2018.

[10] nerophung, "github," 27 May 2020. [Online]. Available: https://nerophung.github.io/2020/05/28/genetic-algorithm.