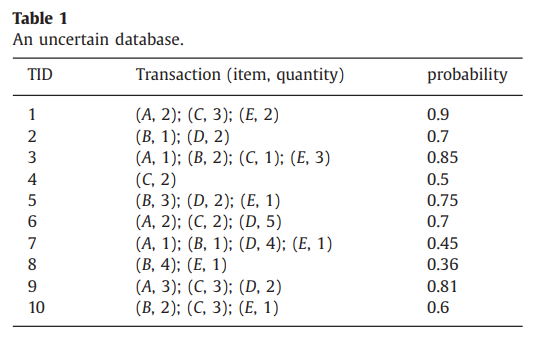
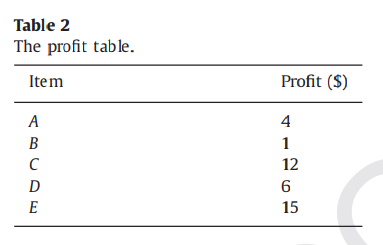
**2. PHUI\_UP**

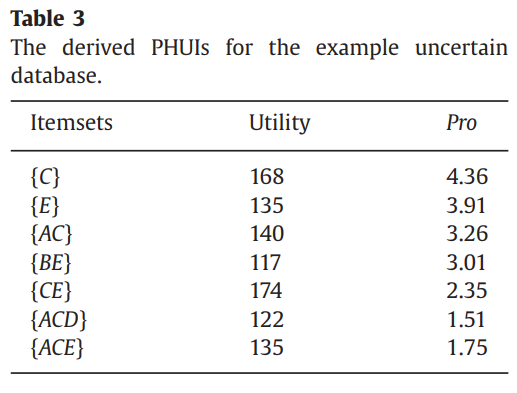
**3.3. Phát biểu bài toán**

Cho một cơ sở dữ liệu không chắc chắn D, tổng độ hữu dụng TU, một ngưỡng hữu dụng tối thiểu ε do người dùng chỉ định, một ngưỡng xác suất tiềm năng tối thiểu μ do người dùng chỉ định, bài toán khai thác tập mục hữu dụng tiềm năng cao (PHUIM) trong cơ sở dữ liệu không chắc chắn là khai thác các tập mục hữu dụng tiềm năng cao (PHUI) có độ hữu dụng lớn hơn hoặc bằng (ε × TU), và xác suất tiềm năng của nó lớn hơn hoặc bằng (μ × |D|).

Từ ví dụ, chạy được đưa ra trong Bảng 1 và Bảng 2, tập hợp các PHUI được hiển thị trong Bảng 3 khi ngưỡng hữu dụng tối thiểu được đặt là 25% và ngưỡng xác suất tiềm năng tối thiểu được đặt là 15%.







**4. Các thuật toán được đề xuất để khai thác tập mục hữu dụng tiềm năng cao trong cơ sở dữ liệu không chắc chắn**

Trong phần này, hai thuật toán hiệu quả để khai thác PHUI trong cơ sở dữ liệu không chắc chắn được trình bày, có tên là PHUI-UP và PHUI-List. Thuật toán PHUI-UP được đề xuất đầu tiên như một thuật toán cơ bản để khai thác PHUI bằng cách tiếp cận theo từng mức trong cơ sở dữ liệu không chắc chắn dựa trên phương pháp Apriori-like [7] và mô hình cận trên đã được phát triển. PHUI-List được thiết kế như một thuật toán hiệu quả hơn để khám phá PHUI trực tiếp trong cơ sở dữ liệu không chắc chắn (không cần tạo ứng viên) dựa trên cấu trúc danh sách xác suất-hữu dụng (PU-list) và cây liệt kê tập hợp.

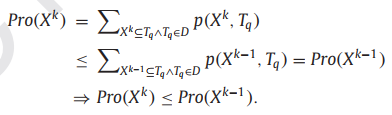
**4.1. Thuật toán PHUI-UP được đề xuất**

Theo hiểu biết tốt nhất của chúng tôi, đây là bài báo đầu tiên thảo luận về PHUIM trong cơ sở dữ liệu không chắc chắn. Một thuật toán PHUI-UP đơn giản được trình bày ở đây để khai thác PHUI theo cách cấp độ dựa trên phương pháp Apriori nổi tiếng [6,7].

**4.1.1. Chiến lược cắt tỉa dựa trên tính chất đóng theo chiều giảm**

Trong thuật toán Apriori nổi tiếng, tính chất đóng theo chiều giảm của độ hỗ trợ được sử dụng để giảm số lượng ứng viên trong bài toán khai thác luật kết hợp (ARM). Tương tự, tính chất đóng theo chiều giảm của thước đo xác suất được đề xuất cũng được sử dụng trong các thuật toán PHUI-UP để khai thác các tập mục hữu dụng tiềm năng cao (PHUI). Chi tiết như sau:

**Định lý 1 (Tính chất đóng theo chiều giảm của tập mục có xác suất cao):** Nếu một tập mục là tập mục có xác suất cao trong cơ sở dữ liệu không chắc chắn, thì tập mục này thỏa mãn tính chất đóng theo chiều giảm trong cơ sở dữ liệu đó.

**Chứng minh.** Gọi Xk là một tập k mục, và Xk-1 là một tập con của nó. Vì p(Xk, Tq) = p(Tq) với bất kỳ giao dịch Tq nào trong cơ sở dữ liệu không chắc chắn D, có thể thấy rằng  
.  
Vì Xk-1 là một tập con của Xk, tập các TID của Xk là tập con của các TID của Xk-1. Do đó,  
  
Nếu Xk là một HPI, xác suất tiềm năng của nó lớn hơn hoặc bằng ngưỡng xác suất tiềm năng tối thiểu, nghĩa là Pro(Xk) ≥ |D| × μ, và do đó Pro(Xk-1) ≥ Pro(Xk) ≥ |D| × μ.

**Hệ quả 1.** Nếu một tập mục Xk là một HPI, mọi tập con Xk-1 của Xk cũng là một HPI.

**Hệ quả 2.** Nếu một tập mục Xk không phải là một HPI, không có tập siêu Xk+1 nào của Xk là một HPI.

Trong bài toán khai thác tập mục hữu dụng cao (HUIM), tính chất đóng theo chiều giảm của ARM không thể được mở rộng trực tiếp để khai thác các tập mục hữu dụng cao (HUI). Tính chất TWDC đã được đề xuất để giảm không gian tìm kiếm trong HUIM.

**Định nghĩa 11.** Độ hữu dụng có trọng số giao dịch (TWU) của một tập mục X trong D là tổng tất cả các độ hữu dụng giao dịch tu(Tq) chứa X, được định nghĩa là  


**Định nghĩa 12.** Một tập mục X trong D được coi là tập mục có độ hữu dụng giao dịch trọng số cao (HTWUI) nếu TWU(X) ≥ TU × ε.

Ví dụ, trong Bảng 1, TWU của mục {E} được tính như sau:  
TWU(E)=tu(T1)+tu(T3)+tu(T5)+tu(T7)+tu(T8)+tu(T10)=74+63+30+44+19+53=283  
Mục {E} là một HTWUI vì TWU(E) (= 283) lớn hơn 134.

Tính chất TWDC cũng được mở rộng để khai thác PHUI trong cơ sở dữ liệu không chắc chắn, và một tính chất mới là đóng theo chiều giảm xác suất-hữu dụng có trọng số giao dịch (TWPUDC) được thiết kế để giảm không gian tìm kiếm của thuật toán PHUI-UP trong việc khai thác PHUI.

**Định nghĩa 13.** Một tập mục X trong cơ sở dữ liệu D được định nghĩa là một tập mục có độ hữu dụng và xác suất trọng số giao dịch cao (HTWPUI) nếu:

1. TWU(X)≥ϵ×TU và
2. Pro(X)≥μ×∣D∣.

Ví dụ, trong Bảng 1 và 2, với μ được đặt là 15%, xác suất tiềm năng tối thiểu được tính là 15%×10=1.5. Xét mục {E} làm ví dụ. Theo Định nghĩa 12, TWU(E)=283, lớn hơn 134. Ngoài ra, Pro(E)=p(E,T1)+p(E,T3)+p(E,T5)+p(E,T7)+p(E,T8)+p(E,T10)=0.9+0.85+0.75+0.45+0.36+0.6=3.91 , lớn hơn 1.51 . Do đó, mục {E} là một HTWPUI.

**Định lý 2 (Tính chất đóng theo chiều giảm của HTWPUI).** Gọi Xk và Xk−1​ là các HTWPUI trong cơ sở dữ liệu không chắc chắn, sao cho Xk−1⊆Xk​. Khi đó, tính chất đóng theo chiều giảm của HTWPUI chỉ ra rằng TWU(Xk−1)≥TWU(Xk) và Pro(Xk−1)≥Pro(Xk).

**Hệ quả 3.** Nếu một tập mục Xk ​ là một HTWPUI, mọi tập con Xk−1 ​của Xk cũng là một HTWPUI.

**Hệ quả 4.** Nếu một tập mục Xk​ không phải là một HTWPUI, không có tập siêu nào Xk+1​ của Xk​ là một HTWPUI.

**Định lý 3 (PHUIs ⊆ HTWPUIs).** Tính chất đóng theo chiều giảm của xác suất và hữu dụng có trọng số giao dịch (TWPUDC) đảm bảo rằng PHUI nằm trong tập HTWPUI. Điều này chỉ ra rằng nếu một tập mục không phải là HTWPUI, thì không có tập siêu nào của nó là PHUI.

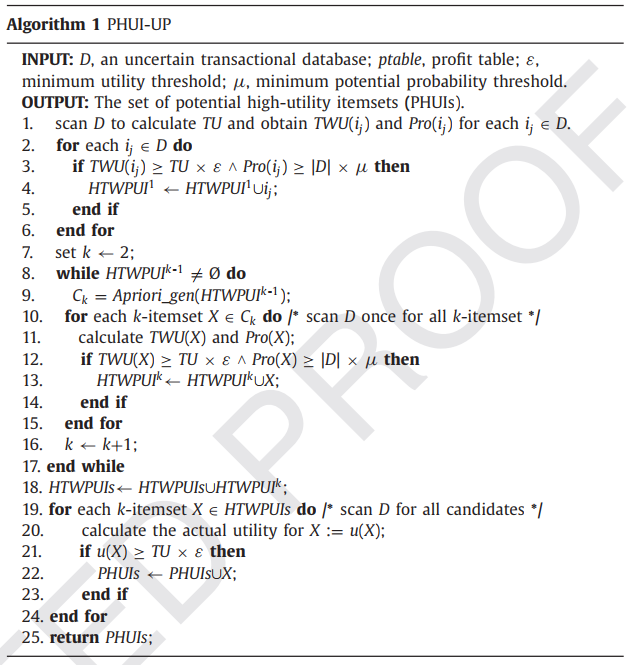
Dựa trên các Định nghĩa 10 và 13, nếu Xk−1​ không phải là HTWPUI, điều đó có nghĩa là Xk−1 không thỏa mãn hai điều kiện:

1. TWU(Xk−1)≥ϵ×TU, và
2. Pro(Xk−1)≥μ×∣D∣; do đó Xk−1​ sẽ không phải là PHUI. Vì vậy, Xk−1 và bất kỳ tập siêu nào của nó Xk​ là HTWPUI, và chúng cũng sẽ không phải là PHUI. Do đó, định lý này đúng.

### **4.1.2. Thuật toán PHUI-UP**

Thuật toán PHUI-UP được đề xuất bao gồm hai giai đoạn: trong giai đoạn đầu tiên, các **HTWPUI** (tập mục có xác suất và hữu dụng trọng số giao dịch cao) được tìm ra, và trong giai đoạn thứ hai, các **PHUI** (tập mục có hữu dụng cao tiềm năng) được trích xuất với một lần quét cơ sở dữ liệu bổ sung. Tính chất **TWPUDC** kế thừa tính chất **TWDC** của mô hình Hai Giai Đoạn (Two-Phase model) để bảo toàn tính chất đóng theo chiều giảm, do đó giảm không gian tìm kiếm cho việc tìm các PHUI. Chỉ những **HTWPUI** còn lại của cấp k−1 mới được sử dụng để tạo ra ứng viên tiếp theo Ck​ ở mỗi cấp.

Dựa trên tính chất **TWPUDC** được thiết kế, **Định lý 3** đảm bảo rằng thuật toán **PHUI-UP** có thể đảm bảo rằng không có tập siêu của các tập mục không phải là HTWPUI nằm trong tập ứng viên sơ bộ (tính đúng đắn) và nó có thể trích xuất tập đầy đủ các PHUI từ tập ứng viên HTWPUI đã được tìm ra (tính đầy đủ). Do đó, thuật toán **PHUI-UP** được đề xuất là đúng đắn và đầy đủ. Dựa trên các định nghĩa và định lý trên, chi tiết của thuật toán **PHUI-UP** được đưa ra như sau.



Như thể hiện trong **Thuật toán 1**, trước tiên thuật toán **PHUI-UP** quét cơ sở dữ liệu để tìm giá trị **TU**, **TWU** và xác suất của tất cả các tập mục đơn trong cơ sở dữ liệu (Dòng 1). Theo các điều kiện thiết kế của **HTWPUI\_k** (Dòng 3), các **HTWPUI\_k** (với k ban đầu được đặt là 1) sau đó được tạo ra (Dòng 2 đến 6) và sẽ tiếp tục được sử dụng để tạo ra các ứng viên tiếp theo Ck+1​ để tìm **HTWPUI\_k+1** theo từng cấp (Dòng 8 đến 17). Trong quá trình này, cơ sở dữ liệu ban đầu phải được quét lại để tìm tập hợp **HTWPUI\_k+1** trong mỗi cấp Ck+1 (Dòng 10). Giai đoạn đầu tiên của **PHUI-UP** kết thúc khi không có ứng viên nào được tạo ra. Sau đó, thuật toán thực hiện giai đoạn thứ hai. Trong giai đoạn này, cần quét lại cơ sở dữ liệu để tìm các **PHUI** cuối cùng từ tập **HTWPUI** (Dòng 19 đến 24).

### Phân tích độ phức tạp

Giả sử m là số lượng giao dịch trong cơ sở dữ liệu D, và n là số lượng mục trong giao dịch lớn nhất trong DDD, và Lk ​ là tập hợp các **HTWPUI\_k**. Bước đầu tiên của thuật toán **PHUI-UP** là tính toán **TU**, **TWU(i)** và giá trị **Pro(i)** của mỗi mục i trong D. Việc này đòi hỏi một lần quét cơ sở dữ liệu, do đó thời gian là O(m×n). Sau đó, cho mỗi cấp độ Ck, các ứng viên được tạo ra bằng cách áp dụng thủ tục con **Apriori\_gen** [7]. Thủ tục này kết hợp các cặp tập mục trong Lk−1​ để tạo ra các tập mục trong Ck​ trong thời gian O(∣Lk∣×∣Lk∣). Sau đó, thuật toán quét toàn bộ cơ sở dữ liệu một lần để tính toán thông tin của mỗi tập mục k trong Ck​. Điều này được thực hiện bằng cách đọc cơ sở dữ liệu một lần và so sánh mỗi tập mục trong Ck​ với mỗi giao dịch, cần thời gian O(m×n×∣Ck∣). Thuật toán kết thúc giai đoạn đầu tiên khi một cấp độ không còn tập nào, và trong trường hợp xấu nhất, có nnn cấp độ. Cuối cùng, cơ sở dữ liệu được quét lại một lần nữa để tính toán chính xác độ hữu dụng của tất cả các **HTWPUI**, trong thời gian O(m×n).

Do đó, trong trường hợp xấu nhất, độ phức tạp thời gian tổng thể của thuật toán **PHUI-UP** là

O(m × n + Pk≥n (|Lk| × |Lk| + m × n × |Ck|) + m × 544 n × |HTWPUIs|)time

Vì phương pháp này sử dụng cách tiếp cận dựa trên **Apriori** để tạo ra các ứng viên ở mỗi cấp độ, nó có thể gặp phải độ phức tạp về không gian và thời gian cao.

### **4.1.3. Ví dụ về thuật toán PHUI-UP**

Để đảm bảo tính nhất quán, hãy xem xét ví dụ chạy trong **Bảng 1** và **Bảng 2**. Giả sử ngưỡng hữu dụng tối thiểu được đặt là 25% và ngưỡng xác suất tiềm năng tối thiểu được đặt là 15%. Do đó, số lượng hỗ trợ tối thiểu và xác suất tiềm năng tối thiểu lần lượt được tính là (442 × 25%) (= 110,5) và (10 × 15%) (= 1,5). Thuật toán **PHUI-UP** được áp dụng như sau.

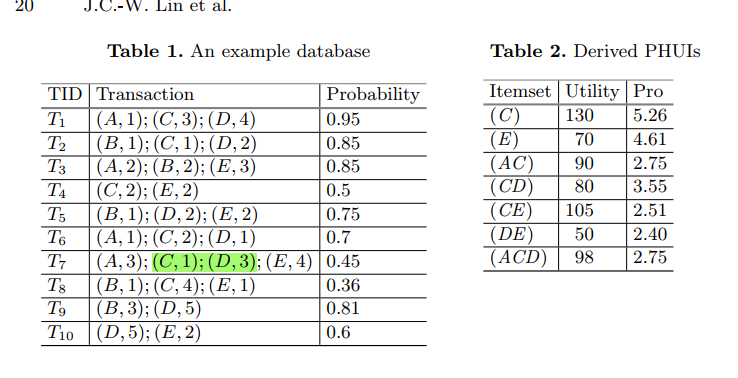
Thuật toán **PHUI-UP** đầu tiên quét cơ sở dữ liệu để tìm giá trị **TWU** và xác suất của tất cả các tập mục đơn trong cơ sở dữ liệu. Kết quả là {A: 303, 3.71; B: 222, 3.71; C: 336, 4.36; D: 209, 3.41; E: 283, 3.91}, trong đó {A: 303, 3.71} chỉ ra rằng giá trị **TWU** của tập mục {A} là 303 và giá trị xác suất của nó là 3.71. Vì tất cả các tập mục đơn đều thỏa mãn điều kiện **TWU(X) ≥ 110,5** và **Pro(X) ≥ 1,5**, nên tất cả chúng được đưa vào tập các tập mục có tiềm năng hữu dụng trọng số giao dịch cao (**HTWPUI**). Sau đó, kkk được đặt là 2, và thuật toán **PHUI-UP** tạo ra các ứng viên C2​ bằng cách áp dụng hàm **Apriori\_gen(HTWPUI1)** theo thứ tự chữ cái của các mục. Tập C2​ là {AB; AC; AD; AE; BC; BD; BE; CD; CE; DE}. Sau đó, cơ sở dữ liệu không chắc chắn được quét lại để tính toán các giá trị **TWU** và **Pro** của mỗi tập mục trong C2​. Kết quả là {AB: 107, 1.3; AC: 259, 3.26; AD: 122, 1.51; AE: 181, 2.2; BC: 116, 1.45; BD: 87, 1.9; BE: 190, 2.05; CD: 122, 1.51; CE: 190, 2.35; DE: 74, 1.2}. Trong số này, chỉ có các tập mục {AC: 259, 3.26; AD: 122, 1.51; AE: 181, 2.2; BE: 190, 2.05; CD: 122, 1.51; CE: 190, 2.35} thỏa mãn điều kiện **TWU(X) ≥ 110,5** và **Pro(X) ≥ 1,5**. Do đó, các tập mục này được thêm vào tập **HTWPUI**.

Sau đó, kkk được đặt là 3 và quy trình này được áp dụng tương tự. Giai đoạn đầu tiên kết thúc khi không có ứng viên nào được tạo ra. Sau giai đoạn đầu tiên, tập đầy đủ các **HTWPUI** được hiển thị trong **Bảng 4**.

Sau đó, thuật toán **PHUI-UP** thực hiện giai đoạn thứ hai. Một lần quét cơ sở dữ liệu bổ sung được thực hiện để tìm các **PHUI** cuối cùng trong tập **HTWPUI** bằng cách tính toán giá trị hữu dụng thực tế của chúng. Hãy xem xét các tập mục {A} và {AC} trong **Bảng 4** làm ví dụ. Theo định nghĩa 4, **u(A) = 36 < 110,5** và **u(AC) = 140 > 110,5**, vì vậy tập mục {A} không phải là **PHUI**, trong khi tập mục {AC} được coi là **PHUI**. Cuối cùng, tập đầy đủ các **PHUI** được phát hiện bởi thuật toán **PHUI-UP**. Kết quả được hiển thị trong **Bảng 3**.

## **4.1 Cấu trúc PU-list**

Cấu trúc PU-list [13] là một cấu trúc dữ liệu theo chiều dọc mới, nó tích hợp các thuộc tính xác suất và tiện ích để giữ lại thông tin cần thiết từ dữ liệu không chắc chắn về mặt thông tin TID, xác suất, tiện ích và thông tin tiện ích còn lại. Giả sử một tập hợp mục X và một giao dịch (hoặc tập hợp mục) T sao cho X⊆T, **tập hợp tất cả các mục từ T không có trong X được ký hiệu là T∖X**, và **tập hợp tất cả các mục xuất hiện sau X trong T được ký hiệu là T/X**. Do đó, T/X⊆T∖X

  
Ví dụ, xét X={CD} và giao dịch T7 trong Bảng 1, T7∖X={AE}, và T7/X={E}.

**3**

**Định nghĩa 16**: **HTWUI** Một tập hợp mục X trong một cơ sở dữ liệu D được gọi là tập hợp mục có tiện ích trọng số giao dịch cao (HTWUI - high transaction-weighted utilization itemset) nếu TWU(X)≥TU×ε.

**Định nghĩa 17**: **HTWPUI** Một tập hợp mục X trong một cơ sở dữ liệu D được gọi là tập hợp mục có xác suất và tiện ích trọng số giao dịch cao (HTWPUI - high transaction-weighted probabilistic and utilization itemset) nếu (1) TWU(X) ≥ ε×TU, và (2) Pro(X) ≥ μ×∣D

**Định lý 1** (Thuộc tính đóng xuống của HTWPUI): Giả sử Xk−1 là một(k−1)​ itemset. Hơn nữa, giả sử Xk là một k-itemset và là một tập siêu tập của Xk−1 (tức là Xk−1⊆Xk ​), và giả sử rằng cả Xk và Xk−1 đều là HTWPUIs trong cơ sở dữ liệu không chắc chắn. Thuộc tính đóng xuống của HTWPUIs cho thấy rằng TWU(Xk−1) ≥ TWU(Xk) và Pro(Xk−1) ≥ Pro(Xk) (Lin et al. 2015c).

**Định lý 2** (PHUIs ⊆ HTWPUIs): Thuộc tính đóng xuống xác suất và tiện ích có trọng số giao dịch (TWPUDC) đảm bảo rằng PHUIs⊆. Do đó, nếu một tập hợp mục không phải là HTWPUI, thì không có tập siêu tập nào của nó là PHUIs (Lin et al. 2015c).

**Chứng minh**: Theo Lin et al. (2015c), định lý này đúng.

***Chiến lược cắt tỉa 1***: Trong lần quét cơ sở dữ liệu đầu tiên, chúng ta có thể thu được giá trị TWU và xác suất của mỗi mục xuất hiện trong cơ sở dữ liệu. Nếu TWU của một mục iii (TWU(i)) và tổng của tất cả các xác suất của iii (Pro(i)) không thỏa mãn hai điều kiện: 1) TWU(i) ≥ ε × TU, và 2) Pro(i) ≥ μ × |D|, thì mục này có thể được cắt tỉa trực tiếp, vì không có tập siêu tập nào của nó là PHUI.

**Cơ sở lý luận**: Theo Định lý 1 và 2, chiến lược cắt tỉa này là hợp lý.

***Chiến lược cắt tỉa 2***: Khi duyệt cây liệt kê tập hợp bằng cách sử dụng tìm kiếm theo chiều sâu, nếu tổng của tất cả các xác suất của một nút cây trong PU-list được xây dựng của nó nhỏ hơn xác suất tiềm năng tối thiểu, thì không có đứa con nào của nó là PHUI.

**Cơ sở lý luận**: Lemma 1 cho thấy rằng nếu tổng xác suất của một tập hợp mục nhỏ hơn xác suất tiềm năng tối thiểu (μ × |D|), thì tập hợp mục này không thể là PHUI, cũng như các nút con (các phần mở rộng) của nó. Do đó, bằng cách tính tổng tất cả các xác suất của mỗi tập hợp mục, nhiều tập hợp mục không hứa hẹn và các phần mở rộng của chúng có thể được xác định là không liên quan và cắt tỉa trực tiếp.

***Chiến lược cắt tỉa 3:*** Khi duyệt cây liệt kê tập hợp bằng cách sử dụng chiến lược tìm kiếm theo chiều sâu, nếu tổng của X.IU và X.RU trong PU-list được xây dựng nhỏ hơn số lượng tiện ích tối thiểu, thì không có đứa con nào (các phần mở rộng) của nút X là PHUI.

**Cơ sở lý luận**: Lemma 2 cho thấy rằng nếu tổng X.IU+X.RUX.IU + X.RUX.IU+X.RU của một nút cây nhỏ hơn số lượng tiện ích tối thiểu (ε × TU), thì bất kỳ nút con nào của nó (các phần mở rộng) là một tập hợp mục không hứa hẹn (không phải là PHUI). Do đó, bằng cách tính tổng tiện ích và tiện ích còn lại cho mỗi nút, các nút con của các tập hợp mục không hứa hẹn có thể được xác định là không liên quan và được cắt tỉa trực tiếp.

Hơn nữa, chiến lược cắt tỉa đồng xuất hiện tiện ích ước tính (EUCP) hiệu quả (Fournier-Viger et al. 2014) cũng được mở rộng trong thuật toán MUHUI được đề xuất để tăng tốc độ phát hiện PHUIs. EUCP được định nghĩa như sau.

***Chiến lược cắt tỉa 4:*** Giả sử X là một tập hợp mục (nút) được gặp trong quá trình tìm kiếm theo chiều sâu của cây liệt kê tập hợp. Nếu TWU của một tập hợp 2 mục Y⊆XY  theo ***EUCS*** được xây dựng nhỏ hơn ngưỡng tiện ích tối thiểu, thì X không phải là HTWPUI và nó cũng không phải là PHUI. Do đó, không có nút con nào của nó là PHUI, và việc xây dựng PU-lists của X và các con của nó không cần phải thực hiện.

**Cơ sở lý luận**: Theo Định nghĩa 17, Lemmas 1 và 3, chiến lược cắt tỉa này là chính xác. Vì PHUIs⊆HTWPUIsPHUIs \subseteq HTWPUIsPHUIs⊆HTWPUIs, nếu TWU(X)≤TU×εTWU(X) ≤ TU × εTWU(X)≤TU×ε cho một tập hợp 2 mục XXX, thì XXX không phải là HTWUI. Hơn nữa, XXX và tất cả các tập siêu tập của nó không phải là PHUIs (lưu ý rằng tất cả các phần mở rộng của XXX trong cây liệt kê tập hợp là các tập siêu tập của XXX). Dựa trên chiến lược EUCP, một số lượng lớn các tập hợp mục không hứa hẹn kkk (k ≥ 2) có thể được cắt tỉa.

Vì thông tin về xác suất của một tập hợp mục X (Pro(X)) có thể dễ dàng thu được từ PU-list của nó, Pro(X) cũng có thể được sử dụng để giảm không gian tìm kiếm. Ý tưởng chính là sau khi xây dựng PU-list của một tập hợp mục XXX, nếu X.PUL rỗng hoặc nếu Pro(X)<∣D∣×μPro(X) < |D| × μPro(X)<∣D∣×μ, thì PU-list X.PULX.PULX.PUL có thể được bỏ qua trực tiếp, và không cần phải thêm vào tập hợp các phần mở rộng của nút cha của XXX. Do đó, X.PULX.PULX.PUL sẽ không được sử dụng để tạo ra bất kỳ tập hợp mục nào khác.

***Chiến lược cắt tỉa 5:*** Giả sử X là một tập hợp mục (nút) được gặp trong quá trình tìm kiếm theo chiều sâu của cây liệt kê tập hợp. Sau khi xây dựng PU-list của tập hợp mục X, nếu X.PUL rỗng hoặc Pro(X) nhỏ hơn ngưỡng xác suất tối thiểu, thì X không phải là PHUI, và không có nút con nào của nó là PHUI. Do đó, PU-lists của các nút con của X không cần phải được xây dựng.

**Cơ sở lý luận**: Theo Lemma 1, chiến lược cắt tỉa này là chính xác.

Dựa trên năm chiến lược cắt tỉa được thiết kế, một số lượng lớn các tập hợp mục không hứa hẹn có thể được cắt tỉa hiệu quả và việc xây dựng PU-lists của các phần mở rộng của chúng có thể được tránh, điều này hiệu quả giảm số lượng các thao tác kết hợp và không gian tìm kiếm trong cây liệt kê tập hợp. Thuật toán MUHUI được đề xuất được mô tả tiếp theo.

**4**

**Định lý 1.** Tổng xác suất của tất cả các nút trong cây liệt kê tập hợp luôn lớn hơn hoặc bằng tổng xác suất của bất kỳ nút con nào của nó.

**Định lý 2.** Đối với bất kỳ nút nào X trong cây liệt kê tập hợp, tổng của X.IU và X.RU lớn hơn hoặc bằng tổng của tất cả các tiện ích của bất kỳ nút con nào của nó.

**Các chiến lược cắt tỉa được đề xuất**  
Dựa trên cấu trúc PU-list và các thuộc tính của xác suất và tiện ích, năm chiến lược cắt tỉa hiệu quả được thiết kế trong MUHUI để loại bỏ sớm các tập mục không có tiềm năng.

**Định lý 1 (Tính chất đóng xuống của HTWPUI).** Giả sử Xk ​ và Xk−1​ là các HTWPUI từ các cơ sở dữ liệu không chắc chắn, và Xk−1⊆X​. Ta có TWU(Xk−1)≥TWU(Xk) và Pro(Xk−1)≥Pro(Xk).  
**Chứng minh.** …

**Định lý 2 (PHUIs ⊆ HTWPUIs).** Tính chất đóng xuống xác suất và tiện ích theo trọng số giao dịch (TWPUDC) đảm bảo rằng PHUIs⊆HTWPUIs, điều này có nghĩa là nếu một tập hợp mục không phải là HTWPUI, thì không có tập siêu tập nào của nó sẽ là PHUI [13].  
Bằng cách sử dụng tính chất TWPUDC, chúng ta chỉ cần xây dựng PU-list cho những tập hợp mục có tiềm năng liên quan đến các HTWPUIs. Hơn nữa, các chiến lược cắt tỉa sau đây được đề xuất trong MUHUI để tăng tốc các phép tính.

**Chiến lược 1.** Sau lần quét cơ sở dữ liệu đầu tiên, chúng ta có thể thu được giá trị TWU và xác suất của mỗi mục 1 trong cơ sở dữ liệu. Nếu TWU của một mục 1 i (TWU(i)) và tổng xác suất của i (Pro(i)) không thỏa mãn hai điều kiện của HTWPUI, thì mục này có thể bị loại bỏ trực tiếp, và không có siêu tập nào của nó là PHUI mong muốn.

**Chiến lược 2.** Khi duyệt qua cây liệt kê tập hợp dựa trên chiến lược tìm kiếm theo chiều sâu, nếu tổng xác suất của một nút cây X liên quan đến Pro(X) trong PU-list đã được xây dựng của nó nhỏ hơn xác suất tiềm năng tối thiểu, thì không có nút con nào của nút này là PHUI mong muốn.

**Chiến lược 3.** Khi duyệt qua cây liệt kê tập hợp theo chiến lược tìm kiếm theo chiều sâu, nếu ***tổng của X.IU và X.RU*** của bất kỳ nút X nào nhỏ hơn số lượng tiện ích tối thiểu, bất kỳ nút con nào của nó đều không phải là PHUI, chúng có thể được coi là không liên quan và sẽ bị loại bỏ trực tiếp.

**Định lý 3 (Chiến lược cắt tỉa đồng thời ước tính tiện ích, EUCP).** Nếu TWU của một tập hợp mục 2 nhỏ hơn số lượng tiện ích tối thiểu, thì bất kỳ siêu tập nào của tập hợp mục 2 này cũng không phải là HTWUI và sẽ không phải là HUI [17].  
Để áp dụng hiệu quả chiến lược EUCP, một cấu trúc được gọi là Cấu trúc Đồng thời ước tính tiện ích (EUCS) [17] được xây dựng trong thuật toán được đề xuất. Đây là một ma trận lưu trữ các giá trị TWU của các tập hợp mục 2, như được hiển thị trong Hình 2 (bên phải). Lưu ý rằng EUCS được xây dựng trong lần quét cơ sở dữ liệu đầu tiên sau khi phát hiện HTWPUI1.

**Chiến lược 4.** Giả sử X là một tập hợp mục (nút) được gặp trong quá trình tìm kiếm theo chiều sâu của cây liệt kê tập hợp. Nếu TWU của một tập hợp mục 2 Y⊆X theo EUCS đã được xây dựng nhỏ hơn ngưỡng tiện ích tối thiểu, thì X không phải là HTWPUI và sẽ không phải là PHUI; không có nút con nào của nó là PHUI. Việc xây dựng PU-lists của X và các nút con của nó là không cần thiết.

**Chiến lược 5.** Giả sử X là một tập hợp mục (nút) được gặp trong quá trình tìm kiếm theo chiều sâu của cây liệt kê tập hợp. Sau khi xây dựng PU-list của một tập hợp mục, nếu X.PUL rỗng hoặc giá trị Pro(X) nhỏ hơn ngưỡng xác suất tối thiểu, thì X không phải là PHUI, và không có nút con nào của X là PHUI. Việc xây dựng PU-lists cho các nút con của X là không cần thiết.

