

Created with an evaluation copy of Aspose.Words. To remove all limitations,
you can use Free Temporary License
<https://products.aspose.com/words/temporary-license/>

KHAI THÁC CÁC TẬP MỤC CÓ LỢI ÍCH CAO SỬ DỤNG TỐI ƯU HÓA ĐÀN KIẾN

Phan Hiếu Minh Tâm – Nguyễn Thảo Vy – Trần Khánh Duyên

Khoa Công nghệ thông tin- Trường Đại học Công nghệ Tp. HCM

Tóm tắt- Khai thác các tập mục có lợi ích cao (HUIs) là một trong những chủ đề nghiên cứu quan trọng nhất trong lĩnh vực khai phá dữ liệu. Các thuật toán dựa trên tính toán tiến hóa đang thu hút ngày càng nhiều sự chú ý, nhờ khả năng tránh sự bùng nổ tổ hợp của không gian tìm kiếm HUI. Trong bài báo này, chúng tôi đề xuất một thuật toán khai thác HUI dựa trên tối ưu hóa đàn kiến (HUIM-ACO), tạo ra các ứng viên mới theo cách xây dựng. Trong thuật toán HUIM-ACO, sử dụng một đường tìm kiếm để đại diện cho các ứng viên HUI. Mỗi đường tìm kiếm được khởi tạo theo tỷ lệ và được mở rộng thêm từng mục theo mỗi bước. Ma trận pheromone được đề xuất để lưu trữ giá trị pheromone của một cặp mục, cho phép cập nhật giá trị cục bộ và toàn cục. Hơn nữa, một cấu trúc quad đường kiến (ant-path quad) được thiết kế để liệt kê hiệu quả các HUI. So sánh thực nghiệm với ba thuật toán khai thác HUI hiện đại nhất cho thấy HUIM-ACO có thể khám phá nhiều HUI hơn trong thời gian ngắn hơn.

Từ khóa: Khai phá dữ liệu · Tập mục có lợi ích cao · Tối ưu hóa đàn kiến · Ma trận pheromone · Ant-path quad

I. Giới Thiệu

Khai thác tập mục có lợi ích cao (HUIM) đã trở thành chủ đề được quan tâm mạnh mẽ trong lĩnh vực khai phá dữ liệu những năm gần đây. Là một bước phát triển từ các tập mục phổ biến (FIs), HUIs không chỉ tập trung vào tần suất xuất hiện mà còn chú ý đến cả số lượng và lợi nhuận liên quan đến các tập hợp mục. Trong bài toán HUIM, mỗi mục có lợi nhuận riêng và có thể xuất hiện nhiều lần trong một giao dịch. Lợi ích của một tập mục được xác định bằng tổng tích của lợi nhuận mỗi mục với số lần xuất hiện của nó trong các giao dịch liên quan. Mục tiêu chính của HUIM là tìm ra tất cả các tập mục có lợi ích đạt hoặc vượt qua ngưỡng tối thiểu do người dùng chỉ định.

Trong nghiên cứu này, chúng ta sẽ xem xét vấn đề HUIM dưới góc độ của thuật toán tối ưu hóa đàn kiến (ACO) và đề xuất một phương pháp mới mang tên HUIM-ACO. Phương pháp này xây dựng mô hình HUIM dựa trên ACO, bao gồm đường tìm kiếm và pheromone. Sử dụng cấu trúc bitmap trình bày chi tiết cách hoạt động của HUIM-ACO và so sánh nó với ba thuật toán

hiện có, qua đó chứng minh được hiệu suất vượt trội cả về hiệu quả lẫn số lượng HUIs tìm được.

II. Các Công Trình Liên Quan

Dù đã có nhiều thuật toán được đề xuất để giải quyết vấn đề HUIM, việc liệt kê chính xác tất cả các HUIs vẫn dẫn đến không gian tìm kiếm khổng lồ khi số lượng mục hoặc kích thước cơ sở dữ liệu tăng cao. Hơn nữa, trong các ứng dụng thực tiễn như hệ thống gợi ý, không nhất thiết phải sử dụng toàn bộ các HUIs đã tìm được.

Để khắc phục những hạn chế này, các phương pháp tính toán tiến hóa đã được áp dụng để tìm kiếm phần lớn các HUIs trong một thời gian hợp lý. Kannimuthu và Premalatha đã phát triển hai thuật toán HUIM dựa trên thuật toán di truyền (GAs), với và không có thiết lập ngưỡng lợi ích tối thiểu. Tuy nhiên, cả hai thuật toán đều gặp phải vấn đề hội tụ sớm, dẫn đến việc tìm được ít HUIs và thời gian thực thi kéo dài. Lin cùng cộng sự đã đề xuất hai thuật toán HUIM dựa trên tối ưu hóa đàn hạt (PSO), trong đó HUIM-BPSO vượt trội hơn nhờ sử dụng cấu trúc cây sáng tạo. Song và Huang cũng đã áp

dụng thuật toán đàn ong nhân tạo (ABC) để mô hình hóa vấn đề HUIM, tối ưu hóa quy trình bằng cách lần lượt cải tiến các nhóm ong thợ, ong quan sát và ong trình sát thông qua các vòng lặp.

Khác biệt với các phương pháp trên, khung tổng quát Bio-HUIF đã được giới thiệu nhằm duy trì tất cả các HUIs đã tìm được, thay vì chỉ chọn những tập có lợi ích cao nhất. Phương pháp này giúp duy trì sự đa dạng trong quần thể, từ đó phù hợp hơn với đặc điểm của bài toán HUIM.

III. Cơ Sở Lý Thuyết

3.1 Một số khái niệm về HUIM

HUIM (High-Utility Itemset Mining) là một kỹ thuật khai thác dữ liệu nhằm tìm ra các tập hợp mục (itemsets) trong một tập dữ liệu giao dịch mà không chỉ phổ biến (xuất hiện thường xuyên) mà còn có giá trị hoặc lợi ích cao theo một tiêu chí định trước, chẳng hạn như lợi nhuận, chi phí hoặc mức độ quan trọng của từng mục.

Khác với các phương pháp khai phá tập phổ biến truyền thống (Frequent Itemset Mining - FIM), HUIM không chỉ xem xét tần suất xuất hiện của các mục trong giao dịch mà còn tích hợp thông tin về "độ hữu ích" (utility), thường được đo lường thông qua trọng số hoặc giá trị liên quan đến từng mục trong các giao dịch.

Giả sử $I = \{i_1, i_2, \dots, i_m\}$ là một tập hợp hữu hạn các mục và $X \subseteq I$ là một tập mục; một tập mục chứa k mục được gọi là k -itemset. Giả sử $D = \{T_1, T_2, \dots, T_n\}$ là cơ sở dữ liệu giao dịch. Mỗi giao dịch $T_a \in D$ là một tập con của I .

Tiện ích nội bộ $q(i_p, T_a)$ biểu thị số lượng của mục i_p trong giao dịch T_a . Tiện ích bên ngoài $p(i_p)$ là giá trị lợi nhuận đơn vị của mục i_p . Tiện ích của mục i_p trong giao dịch T_a được định nghĩa là $u(i_p, T_a) = p(i_p) \times q(i_p, T_a)$. Tiện ích của tập mục X trong giao dịch T_a được định nghĩa là: $u(X, T_a) = \sum_{i_p \in X} u(i_p, T_a)$. Tiện ích của tập mục X trong D được định nghĩa là $u(X) = \sum_{T_a \in D} u(X, T_a)$. Tiện ích giao dịch của giao dịch T_a được định nghĩa là $TU(T_a) = u(T_a, T_a)$. Ngưỡng tiện ích tối thiểu δ được đưa ra dưới dạng phần trăm của tổng giá trị tiện ích giao

dịch của cơ sở dữ liệu, trong đó giá trị tiện ích tối thiểu được định nghĩa là $\min_util = \delta \times \sum_{T_a \in D} TU(T_a)$. Một tập mục X được gọi là tập mục có tiện ích cao nếu $u(X) \geq \min_util$.

Tiện ích trọng số giao dịch (TWU) của tập mục X [7] là tổng tiện ích giao dịch của tất cả các giao dịch chứa X , được định nghĩa là $TWU(X) = \sum_{T_a \in D} TU(T_a)$. X là một tập mục có tiện ích trọng số giao dịch cao (HTWUI) nếu $TWU(X) \geq \min_util$; ngược lại, X là một tập mục có tiện ích trọng số giao dịch thấp (LTWUI). Một HTWUI/LTWUI với k mục được gọi là k -HTWUI/ k -LTWUI.

Xem xét CSDL giao dịch trong bảng 1 và lợi nhuận trong bảng 2. Để thuận tiện, chúng tôi viết tập mục $\{C, E\}$ là CE . Trong CSDL ví dụ, độ hữu ích của hạng mục E trong giao dịch $T3$ là $u(E, T3) = 3 \times 2 = 6$, độ hữu ích của tập mục CE trong giao dịch $T1$ là $u(CE, T1) = u(C, T1) + u(E, T1) = 3 + 3 = 6$, và độ hữu ích của tập mục CE trong CSDL giao dịch là $u(CE) = u(CE, T1) + u(CE, T0) + u(CE, T3) + u(CE, T4) = 27$. Cho $\min_util = 28$, với $u(CE) < \min_util$, vậy CE không là một HUI. TU của $T0$ là $TU(T0) = u(ABCDE, T0) = 25$, và những độ hữu ích của những giao dịch khác được trình bày trong cột thứ ba của bảng 1. TWU của tập mục CE là $TWU(CE) = TU(T0) + TU(T1) + TU(T3) + TU(T4) = 76$, vậy CE là một HTWUI.

TID	Transaction
T0	(A,1), (B,5), (C,1), (D,3), (E,1)
T1	(B,4), (C,3), (D,3), (E,1)
T2	(A,1), (C,1), (D,1)
T3	(A,2), (C,6), (E,2)
T4	(B,2), (C,2), (E,1)

Bảng 1: Các giao dịch

Item	Price
A	5
B	2
C	1
D	2
E	3

Bảng 2: Lợi nhuận

3.2 Một số khái niệm về ACO

ACO (Ant Colony Optimization) là một thuật toán tiến hóa được lấy cảm hứng từ hành vi tìm kiếm thức ăn của loài kiến, rất phù hợp để giải các bài toán tổ hợp. Do bản chất của HUIM (High-Utility Itemset Mining) là một bài toán tổ hợp, nghiên cứu này đã áp dụng ACO để khám phá hiệu quả các tập hợp mục có lợi ích cao (HUIs).

Trong ACO, vấn đề được giải quyết thông qua việc mô phỏng một số lượng kiến nhân tạo di chuyển trên một đồ thị. Trên đồ thị này, mỗi đỉnh đại diện cho một địa điểm, còn mỗi cạnh biểu thị mối liên kết giữa hai địa điểm. Một biến gọi là pheromone được gắn với mỗi cạnh, và các con kiến có thể đọc hoặc thay đổi giá trị của biến này.

Thuật toán ACO hoạt động theo cách lặp lại và được thực thi bởi SN con kiến. Trong mỗi vòng lặp, từng con kiến At ($1 \leq t \leq SN$) sẽ xây dựng một lời giải bằng cách di chuyển từ đỉnh hiện tại về đến đỉnh tiếp theo trên đồ thị, sao cho vn chưa được At ghé thăm. Khi di chuyển, đỉnh tiếp theo vn được chọn dựa trên xác suất tỷ lệ thuận với lượng pheromone liên kết với cạnh (vc, vn). Sau đó, giá trị pheromone trên cạnh (vc, vn) sẽ được cập nhật, nhằm định hướng để At có xu hướng tạo ra những lời giải tương tự các lời giải tốt nhất đã tìm thấy trước đó trong các vòng lặp sau.

Khi tất cả các con kiến đã tìm được đỉnh tiếp theo và cập nhật giá trị pheromone tại vị trí hiện tại của mình, một bước cập nhật toàn cục sẽ được thực hiện để điều chỉnh giá trị pheromone của hành trình tốt nhất. Khác với việc cập nhật của từng con kiến, bước cập nhật toàn cục sẽ thay đổi giá trị pheromone trên tất cả các hành trình mà các con kiến đã đi qua trong vòng lặp này. Quá trình di chuyển, cập nhật pheromone cục bộ và cập nhật pheromone toàn cục sẽ tiếp tục lặp lại cho đến khi điều kiện dừng được thỏa mãn.

IV. Khái niệm HUIM-ACO

4.1 Ma trận Pheromone

Ma trận Pheromone được sử dụng để ghi lại giá trị Pheromone của từng cặp itemset.

Định Nghĩa 1: Hãy để H là số lượng 1-HTWUIs và để ma trận pheromone PM là một ma trận $H \times H$. Mỗi mục $e_{j,k}$ của PM tương ứng với giá trị pheromone của cặp mục 2-itemset $i_j i_k$. Các mục trên đường chéo chính, tương ứng với giá trị pheromone của một mục với chính nó, được đặt là 0. Đối với các mục khác $e_{j,k}$ với $i_j \neq i_k$ được khởi tạo như sau:

$$e_{j,k} = \begin{cases} \frac{u(i_j i_k)}{2}, & \text{if } TWU(i_j i_k) \geq \text{min_until} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Chúng ta có thể thấy từ Eq. 1 rằng $e_{j,k} = e_{k,j}$, vì vậy PM là một ma trận đối xứng và các mục của PM đối xứng so với đường chéo chính. Do đó, chúng ta chỉ cần lưu trữ một nửa PM so với đường chéo chính.

Đối với giai đoạn cập nhật pheromone cục bộ, mỗi mục của PM được sửa đổi như sau:

$$e_{j,k} = e_{j,k} + u(i_j i_k)^\alpha / L \quad (2)$$

(trong đó $\alpha \geq 0$ là hệ số ảnh hưởng cục bộ để điều chỉnh đóng góp của giá trị tiện ích trong pheromone và $L (L > 1)$ là hệ số bay hơi để làm loãng pheromone).

Trong giai đoạn cập nhật pheromone toàn cầu, ta sử dụng một chiến lược khác so với phương pháp ACO thông thường. Thay vì cập nhật tất cả các đường đi trong một lần lặp, một khi một HUI được phát hiện, thuật toán này chỉ cập nhật các mục trong đường đi đó. Cụ thể, đối với mỗi mục $e_{j,k}$ trong đường đi tạo ra một HUI, việc sửa đổi sau được thực hiện:

$$e_{j,k} = e_{j,k} + TWU(i_j i_k)^\beta / L \quad (3)$$

(trong đó $\beta \geq 0$ là hệ số ảnh hưởng toàn cầu để điều chỉnh đóng góp của giá trị TWU trong pheromone).

Đối với Eqs. 2 và 3, giá trị tiện ích được sử dụng để cập nhật pheromone cục bộ, và giá trị TWU được sử dụng để cập nhật pheromone toàn cầu. Điều này chủ yếu bởi vì mức độ chi tiết của các giá trị tiện ích khác biệt so với TWU tại giai đoạn cập nhật toàn cầu. Vì giai đoạn cục bộ chỉ được thực thi giữa hai đỉnh liền kề, giá trị tiện ích được sử dụng để cập nhật pheromone. Đối với giai đoạn toàn cầu, tất cả các cặp đỉnh liền kề trong đường dẫn hiện tại được cập nhật, vì vậy sử dụng TWU, xấp xỉ của đường dẫn, để cập nhật pheromone.

4.2 Khởi tạo đường đi tìm kiếm

Khi một con kiến bắt đầu đường đi tìm kiếm của nó, nó chọn mục đầu tiên i_j theo tỷ lệ, được xác định bởi:

$$P_j = \frac{TWU(i_j)^\gamma / L}{\sum_{k=1}^H TWU(i_k)^\gamma / L} \quad (4)$$

trong đó γ ($\gamma > 0$) là hệ số ảnh hưởng giao dịch để điều chỉnh đóng góp của giá trị TWU trong việc chọn mục đầu tiên.

Đối với mỗi con kiến A_t , một bộ bốn đường đi của kiến được biểu diễn là $AP_t = \langle i_c, Pa, Succ, TS \rangle$, trong đó i_c là mục hiện tại đang được A_t ghé thăm, Pa là tập hợp các mục đã được A_t ghé thăm, $Succ$ là tập hợp các mục chưa được A_t ghé thăm, và TS là tập hợp các giao dịch chứa tất cả các mục trong Pa .

4.3 Quy tắc chuyển trạng thái

Quy tắc chuyển trạng thái được sử dụng bởi một con kiến để chọn xác suất đỉnh tiếp theo mà nó sẽ ghé thăm. Hãy để A_t là một con kiến. Đỉnh tiếp theo đại diện cho mục i_{next} mà A_t sẽ ghé thăm được xác định bởi:

$$i_{next} = \begin{cases} \underset{i_j \in AP_t.Succ}{argmax} \{e_{c,j}\}, & \text{if } rand \geq \tau \\ i_k \in AP_t.Succ \text{ with a probability } P_{c,k}, & \text{otherwise} \end{cases} \quad (5)$$

trong đó $e_{\{c,j\}}$ là giá trị pheromone giữa $AP_t.i_c$ và i_j trong ma trận pheromone, r là một số ngẫu nhiên phân phối đều giữa 0 và 1, τ ($0 \leq \tau \leq 1$) là ngưỡng lựa chọn được xác định trước bởi người dùng, và xác suất từ mục $AP_t.i_c$ đến mục i_k được định nghĩa là:

$$P_{c,k} = \frac{e_{c,k}}{\sum_{i_l \in AP_t.Succ} e_{c,l}} \quad (6)$$

Công thức này xác định cách xác suất được tính toán dựa trên giá trị pheromone của các cạnh liên quan.

V. Thuật toán HUIM-ACO

5.1 Biểu diễn thông tin Mục bằng Bitmap

Chúng ta sẽ sử dụng bitmap, một cách biểu diễn hiệu quả thông tin mục trong các thuật toán khai thác tập phổ biến (FI mining) và khai thác tập hữu ích cao (HUI mining), trong HUIM-ACO để xác định các giao dịch chứa tập mục tiêu.

Cụ thể, HUIM-ACO sử dụng biểu diễn bitmap cover cho các tập mục. Trong một bitmap cover, mỗi bit đại diện cho một giao dịch trong cơ sở dữ liệu. Nếu mục i xuất hiện trong giao dịch T_j , thì bit j của bitmap cover cho mục i được đặt thành 1; nếu không, bit này được đặt thành 0.

Điều này được mở rộng một cách tự nhiên đến các tập mục. Nếu X là một tập mục, thì $Bit(X)$ tương ứng với bitmap cover đại diện cho tập giao dịch chứa tập mục X .

Giả sử X và Y là hai tập mục. Khi đó, $Bit(X \cup Y)$ có thể được tính bằng $Bit(X) \cap Bit(Y)$, tức là phép toán AND bitwise của $Bit(X)$ và $Bit(Y)$.

5.2 Thuật toán đề xuất

Input:

Cơ sở dữ liệu giao dịch D

Kích thước quần thể SN

Giá trị tiện ích tối thiểu min_util

Yếu tố ảnh hưởng cục bộ α (alpha)

Yếu tố ảnh hưởng toàn cục β (beta)

Hệ số bay hơi ζ (zeta)

Số vòng lặp tối đa max_iter

Output:

Các tập mục có tiện ích cao (HUIs)

Thuật toán 1:

1. **Initialization()**.

2. Gán $H_{new} = \emptyset$ (tập rỗng).

3. Gán $iter = 1$.

4. Lặp lại:
5. For $iter \leq \max_iter$:
6. For mỗi con kiến A_t :
7. If $iter = 1$:
8. Khởi tạo mục đầu tiên mà A_t ghé thăm theo công thức 4.
9. Khởi tạo đường đi của kiến AP_t .
10. End if.
11. If $AP_t.Succ \neq \emptyset$:
12. Chọn mục tiếp theo i_{next} từ vị trí hiện tại AP_t bằng công thức 5.
13. If $TWU(AP_t.Pa \cup i_{next}) \geq \min_util$ và $(AP_t.Pa \cup i_{next} \notin SH)$:
14. Thực hiện **LocalUpdate**(AP_t, i_{next}).
15. If $u(AP_t.Pa) \geq \min_util$:
16. Gán $AP_t.Pa \rightarrow H_{new}$.
17. Thực hiện **GlobalUpdate**(AP_t).
18. End if.
19. End if.
20. End if.
21. Kết thúc vòng lặp con.
22. Tăng $iter++$.
23. Kết thúc vòng lặp chính.
24. Cập nhật $H_{new} \rightarrow SH$.
25. Trong khi $H_{new} \neq \emptyset$.
26. Xuất tất cả các HUIs trong SH.

Trong **Thuật toán 1**, thủ tục Initialization(), được mô tả trong Thuật toán 2, được gọi ở Bước 1. Vòng lặp chính từ Bước 2 đến Bước 25 khai thác các tập hữu ích cao (HUIs) một cách lặp đi lặp lại cho đến khi không còn tập HUI mới nào được phát hiện bởi bất kỳ con kiến nào.

Bước 3 khởi tạo H_{new} , tập hợp các HUIs mới được phát hiện trong một vòng lặp, như một tập rỗng. Bước 4 sau đó khởi tạo số lần lặp là 1.

Vòng lặp từ Bước 5 đến Bước 23 xử lý từng con kiến riêng lẻ với điều kiện rằng thời gian tìm kiếm chưa vượt quá ngưỡng tối đa. Vòng lặp từ Bước 6 đến Bước 21 mô tả các hành động của một con kiến khi cập nhật giá trị pheromone và phát hiện HUIs trong một lần lặp.

Trong lần lặp đầu tiên của một con kiến, mục đầu tiên được chọn và bộ bốn đường đi của kiến được khởi tạo ở Bước 7 đến Bước 10. Nếu vẫn còn các mục để khám phá, mục tiếp theo sẽ được xác định ở Bước 12.

Nếu giá trị TWU của đường đi hiện tại khi thêm mục tiếp theo vừa được xác định không thấp hơn ngưỡng tiện ích tối thiểu và kết quả hợp nhất chưa được phát hiện là một HUI trước đó, thì Bước 14 gọi thủ tục LocalUpdate() được mô tả trong Thuật toán 3.

Tương tự, khi đường đi vừa được hình thành là một HUI, thủ tục GlobalUpdate(), được mô tả trong Thuật toán 4, sẽ được gọi. Tập giao dịch TS trong bộ bốn đường đi của kiến đảm bảo rằng cả giá trị TWU và giá trị tiện ích đều có thể được xác định một cách hiệu quả bằng cách tham chiếu đến các giao dịch trong TS, thay vì toàn bộ cơ sở dữ liệu.

Thời gian tìm kiếm sau đó được tăng lên một đơn vị. Trong Bước 24, các HUIs mới được phát hiện trong vòng lặp đó được thêm vào SH, là tập hợp tất cả các HUIs đã được phát hiện. Cuối cùng, Bước 26 xuất ra tất cả các kết quả khai thác.

Thuật toán 2: Initialization()

1. Quét cơ sở dữ liệu D một lần và xóa các 1-LTWUIs.
2. Biểu diễn cơ sở dữ liệu đã được tổ chức lại dưới dạng bitmap.
3. Khởi tạo từng phần tử của ma trận pheromone PM bằng Công thức 1.
4. $SH = \emptyset$.

Trong **Thuật toán 2**, các 1-LTWUIs được xóa đầu tiên ở Bước 1. Bước 2 sau đó xây dựng biểu diễn bitmap của cơ sở dữ liệu. Giá trị pheromone của từng cặp 1-HTWUIs được khởi tạo ở Bước 3. Cuối cùng, tập hợp tất cả các

HUIs SHSHSH được khởi tạo là một tập rỗng ở Bước 4.

Thuật toán 3: LocalUpdate (AP_t , i-next)

- 1. Cập nhật giá trị pheromone cục bộ trên $e_{c,t-next}$ bằng Công thức 2.
- 2. Gán $i_{next} \rightarrow AP_t.Pa$.
- 3. $AP_t.Succ = AP_t.Succ \setminus i_{next}$.
- 4. Gán $AP_t.ic = i_{next}$.
- 5. Cập nhật $AP_t.TS$.

Trong **Thuật toán 3**, giá trị pheromone tương ứng với $AP_t.ic$ và i_{next} được cập nhật ở Bước 1. Bước 2 thêm i_{next} vào đường đi của con kiến hiện tại. Sau đó, i_{next} được xóa khỏi tập hợp

các mục chưa được thăm trong Bước 3. Bước 4 sử dụng i_{next} làm mục hiện tại mới. Cuối cùng, tập hợp các giao dịch chứa đường đi hiện tại được cập nhật ở Bước 5.

Thuật toán 4: GlobalUpdate (AP_t , i-next)

- 1. Với $l = 1$ đến $|AP_t.Pa| - 1$, thực hiện:
- 2. Cập nhật giá trị pheromone $e_{l,l+1}$ bằng Công thức 3.
- 3. Kết thúc vòng lặp.

Trong **Thuật toán 4**, các giá trị pheromone của từng cặp mục trong đường đi hiện tại được cập nhật tuần tự từ cặp đầu tiên đến cặp thứ $(|AP_t.Pa|-1)$, trong đó $|AP_t.Pa|$ là số lượng mục trong đường đi hiện tại $AP_t.Pa$.

VI. Ví dụ minh họa

Chúng tôi dùng CSDL giao dịch ở bảng 1 và bảng lợi nhuận ở bảng 2 để giải thích. Sau lần quét CSDL đầu tiên, TWU của mỗi hạng mục được biểu diễn trong bảng 3 dưới đây.

Itemset	TWU
A	55
B	54
C	84
D	53
E	76
A,B	25
A,C	55
A,D	33
A,E	47
B,C	54
B,D	45
B,E	54
C,D	53
C,E	76
D,E	45
A,B,C	25
A,B,D	25

This document was truncated here because it was created in the Evaluation Mode.

