

HỘI THẢO KHOA HỌC VÀ CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

BAN TỔ CHỨC

Trưởng ban:

PGS.TS. Đặng Trần Khánh

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

Thành viên:

ThS. Nguyễn Thị Định

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Thê Hữu

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Phan Thị Ngọc Mai

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Văn Tùng

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

BAN CHƯƠNG TRÌNH

Trưởng ban:

PGS.TS. Đặng Trần Khánh

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

Thành viên:

TS. Hoàng Xuân Bách

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

PGS.TS. Nguyễn Tuấn Đăng

Trường ĐH Sài Gòn

ThS. Nguyễn Thị Định

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Ngô Dương Hà

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

TS. Ngô Thanh Hùng

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Văn Lẽ

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

TS. Nguyễn Thanh Long

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

TS. Vũ Đức Lý

Trường Đại học FPT

TS. Nguyễn Thị Bích Ngân

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

TS. Huỳnh Hữu Nghĩa

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

KS. Trần Trương Tuấn Phát

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

TS. Lê Lam Sơn

Trường ĐH Bách Khoa – ĐH Quốc gia Tp.HCM

TS. Vũ Đức Thịnh

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

PGS.TS. Lê Hồng Trang

Trường ĐH Bách Khoa – ĐH Quốc gia Tp.HCM

ThS. Hà Lê Hoài Trung

Trường ĐH Công nghệ Thông tin – ĐH Quốc gia Tp.HCM

BAN THƯ KÝ

Trưởng ban:

ThS. Nguyễn Thị Định

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

Thành viên:

CN. Nguyễn Hải Bình

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Thị Thu Tâm

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Thị Thanh Thủy

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Trần Nhu Ý

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

ThS. Nguyễn Thị Hải Yến

Trường ĐH Công nghiệp Thực phẩm Tp.HCM

MỤC LỤC

1. Những vấn đề bảo mật trong mô hình học cộng tác - *Hà Lê Hoài Trung¹, Đặng Trần Khánh^{2,*}* 1
2. Ứng dụng AI trong thương mại điện tử thuật toán Content Based System hỗ trợ gợi ý mua hàng - *Nguyễn Hải Yến^{1,*}* 19
3. Một phương pháp gom cụm dữ liệu cho bài toán tư vấn lựa chọn học phần tự chọn - *Nguyễn Thị Thuỳ Trang^{1,*}, Phan Thị Ngọc Mai¹* 31
4. Ứng dụng thuật giải tối ưu bầy đàn ẩn tập hữu ích cao nhạy cảm - *Lâm Thị Họa Mi^{1,*}, Vũ Văn Vinh¹, Dương Thị Mộng Thùy¹* 40
5. Khai thác tập hữu ích cao dựa trên cơ chế diffset - *Nguyễn Thị Thanh Thủy^{1,*}, Nguyễn Văn Lẽ¹, Mạnh Thiên Lý¹* 56
6. Mô hình cá nhân hóa người học trên cơ sở logic mờ ứng dụng trong hệ thống học mở - *Hoàng Xuân Bách^{1,*}* 72
7. Phát hiện các đối tượng trong ảnh tế bào máu bằng kỹ thuật học sâu - *Trần Đình Toàn^{1,*}, Nguyễn Đức Toàn²* 76
8. Ứng dụng thuật toán song song FP-Growth khai phá tập phổ biến - *Nguyễn Thị Hồng Thảo^{1,*}* 86
9. Hệ thống chú thích ảnh tự động cho người khiếm thị - *Đinh Thị Mận¹, Nguyễn Văn Thịnh², Trần Ngọc Hiếu¹, Trần Thị Xinh, Trần Thị Vân Anh^{1,*}* 92
10. Ứng dụng máy học vào bài toán dự đoán bệnh ung thư vú - *Huỳnh Thị Châu Lan^{1,*}, Nguyễn Hải Yến¹* 104
11. Xây dựng hệ thống dự đoán kết quả học tập của sinh viên nhằm hỗ trợ công tác cõi ván học tập - *Ngô Thanh Hùng¹, Nguyễn Thị Thu Tâm^{1,*}* 112
12. Nguy cơ tấn công DDoS từ các thiết bị IoT – *Trần Đắc Tốt^{1,*}* 126
13. Nhận diện các video có yếu tố chính trị Việt Nam trên mạng xã hội - *Trần Thị Vân Anh¹, Đặng Trần Khánh^{1,*}, Nguyễn Hải Bình¹* 147
14. Cơ sở lý thuyết các mô hình độ chấp nhận và sử dụng công nghệ - *Đào Minh Châu^{1,*}* 160
15. Nâng cao hiệu quả tìm kiếm ảnh dựa trên cấu trúc KD-Tree và K-Means - *Nguyễn Thị Định^{1,*}, Trần Thị Bích Vân¹, Nguyễn Phương Hạc¹* 169
16. Tiếp cận các phương pháp phân lớp dữ liệu trong dự báo chất lượng nước - *Nguyễn Thị Diệu Hiền¹, Bùi Công Danh^{1,*}* 182

17. Nghiên cứu thuật toán tối ưu đàm kién tiến hóa – *Dinh Nguyễn Trọng Nghĩa¹, Nguyễn Thị Thu Tâm^{1,*}* **190**
18. Tổng quan về học liên kết và tiềm năng ứng dụng trong lĩnh vực y học - *Trần Trương Tuấn Phát^{1,2}, Đặng Trần Khánh^{1,*}, Nguyễn Thị Thanh Thảo¹* **198**
19. Tìm hiểu mô hình GPT và ứng dụng GPT phục vụ cho học tập và nghiên cứu ở sinh viên - *Đinh Huy Hoàng^{1,*}* **216**
20. Điểm danh sinh viên tự động bằng phương pháp nhận diện khuôn mặt - *Trần Thị Vân Anh¹, Đặng Trần Khánh^{1,*}, Nguyễn Vương Triều An¹, Nguyễn Văn Sơn¹, Nguyễn Nguyên Khang¹* **229**
21. Các công cụ nâng cao chất lượng và đánh giá giảng dạy - *Đào Minh Châu^{1,*}* **242**
22. HUFI-STPMS: hệ thống quản lý điểm rèn luyện sinh viên Trường Đại học Công nghiệp Thực phẩm (HUFI) - *Trần Trương Tuấn Phát¹, Đặng Trần Khánh¹, Ngô Thành Hùng^{1,*}, Trần Thị Vân Anh¹, Lê Thị Trà My¹, Đoàn Huỳnh Tuyết Anh¹* **248**
23. Nguồn nhân lực ngành công nghệ thông tin trong xu hướng chuyển đổi số - *Trần Thị Bích Vân^{1,*}* **261**
24. Tối ưu hóa đàm kién và ứng dụng trong bài toán ẩn tập phô biến - *Hồ Thanh Hải¹, Dinh Nguyễn Trọng Nghĩa^{1,*}* **265**
25. Hệ thống thông tin y tế một cách tiếp cận dựa trên thành phần - *Huỳnh Hữu Nghĩa¹, Nguyễn Thị Thu Tâm^{1,*}* **274**
26. Tối ưu hóa tìm kiếm chim hòng hạc và ứng dụng trong bài toán ẩn tập phô biến - *Lê Huỳnh Đức¹, Nguyễn Thị Hàng¹, Dinh Nguyễn Trọng Nghĩa^{1,*}* **288**
27. Tối ưu hóa sói xám và ứng dụng trong bài toán ẩn tập phô biến - *Phạm Nguyễn^{1,*}, Dinh Nguyễn Trọng Nghĩa¹* **303**
28. Xây dựng thuật toán Logistic cho phép phân loại mail và mô hình pipeline trên Spark - *Nguyễn Minh Hiền¹, Nguyễn Trọng Đăng Khoa¹, Nguyễn Văn Tùng¹, Ngô Dương Hà^{1,*}* **316**
29. Ứng dụng kỹ thuật resnet phân loại lá cây thuốc chữa bệnh gan - *Phạm Hoàng Hiệp¹, Trần Như Ý¹, Phan Thị Ngọc Mai^{1,*}* **325**
30. Xây dựng hệ thống xử lý tiền trình khóa luận tốt nghiệp - *Phạm Tấn Thuận¹, Vũ Trần Minh Hoàng¹, Ngô Văn Sơn¹, Nguyễn Ngọc Thạch¹, Trần Văn Thọ^{1,*}* **335**
31. Ứng dụng Firebase Authentication và Firebase Cloud Messaging để xây dựng đăng nhập, đăng ký và gửi thông báo về cho người dùng trong lập trình di động Android - *Lê Minh Kha^{1,*}, Biện Thành Nhựt¹, Nguyễn Hữu Trung¹, Nguyễn Thị Bích Ngân¹* **343**

-
32. Điều hướng Navigation trong thiết kế giao diện lập trình di động Android - *Lê Minh Phát^{l,*}, Lê Tâm Nhu^l, Ngô Thị Kim Ngân^l, Nguyễn Thị Kim Tuyên^l, Nguyễn Thị Bích Ngân^l* **360**
33. Xây dựng RESTful api và gọi API từ ứng dụng trong lập trình di động Android - *Vũ Ngô Đạt^{l,*}, Vũ Văn Vinh^l, Nguyễn Thị Bích Ngân^l* **378**
34. HUFI-ELAB: Hệ thống quản lý phòng thí nghiệm Trường Đại học Công nghiệp Thực phẩm (HUFI) - *Trần Trương Tuấn Phát^l, Đặng Trần Khánh^{l,*}, Tống Đăng Khoa^l, Nguyễn Trọng Nghĩa^l, Lê Trạng Lân^l* **393**
35. Thiết kế cơ chế mô hình học liên kết lấy cảm hứng từ mạng Ethereum 2.0 - *Trần Trương Tuấn Phát^l, Đặng Trần Khánh^{l,*}* **395**

TỐI UU HÓA ĐÀN KIẾN VÀ ỨNG DỤNG TRONG BÀI TOÁN ẨN TẬP PHỔ BIẾN

Hồ Thanh Hải¹, Đinh Nguyễn Trọng Nghĩa^{1,*}

¹Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh

*Email: nghiadnt@hufi.edu.vn

Ngày nhận bài: 27/04/2023; Ngày chấp nhận đăng: 10/05/2023

TÓM TẮT

Bài báo này trình bày về thuật toán tối ưu hóa đàm kién là một thuật toán metaheuristic được lấy cảm hứng từ hành vi tìm kiếm thức ăn của kiến và kiến đực. Điểm mạnh của thuật toán tối ưu hóa đàm kién là khả năng tìm kiếm các lời giải tối ưu trong bài toán tổng quát hóa với kích thước lớn. Ngoài ra, bài báo cũng đề xuất một phương pháp áp dụng tối ưu hóa đàm kién vào bài toán ẩn tập phổ biến. Chính vì những ưu điểm của tối ưu hóa đàm kién có thể giúp tìm được lời giải tối ưu cho bài toán ẩn tập phổ biến, giảm thiểu các hiệu ứng phụ.

Từ khóa: Ant colony optimization, ACO, PPDM, hiding frequent itemset.

1. GIỚI THIỆU

Các thuật toán metaheuristic được sử dụng để tìm một lời giải tối ưu gần đúng cho các bài toán tối ưu khó, trong đó không có một phương pháp chính xác nào có thể giải được trong một khoảng thời gian hợp lý. Một thuật toán metaheuristic được định nghĩa là một thuật toán không phụ thuộc bài toán mà có thể tìm các lời giải gần đúng cho các bài toán khó. Các thuật toán metaheuristic thường được lấy cảm hứng từ tự nhiên và cố gắng giải bài toán bằng cách mô phỏng các hiện tượng vật lý, sinh học, ...

Các thuật toán tiến hóa (EA) là các thuật toán metaheuristic ngẫu nhiên và dựa trên quần thể. EA khác với các thuật toán thông thường khác ở chỗ chúng thực hiện tiến hóa quần thể các lời giải để tiến đến một lời giải tối ưu gần đúng. Thông thường, các EA thực hiện tìm kiếm trong không gian bài toán theo cách sau: một quần thể các các thê (lời giải) được phát sinh một cách ngẫu nhiên ban đầu, sau đó độ thích nghi của chúng được đánh giá dựa trên hàm thích nghi.

Trong các thê hệ tiếp theo, các cá thê tiến hóa theo xu hướng của cá thê tốt nhất và sự trợ giúp của hàm thích nghi. Quá trình này tiếp tục cho đến khi nó đạt được số vòng lặp tối đa hoặc tìm được lời giải gần đúng mong đợi. Một vài thuật toán tiến hóa được biết đến như: Thuật giải di truyền (GA) dựa trên học thuyết tiến hóa của Darwin , Phương pháp tối ưu bầy đàm (PSO) dựa trên việc mô phỏng hành vi kiếm ăn của một đàn chim , Phương pháp tìm hòa âm (Harmony Search - HS) dựa trên phương pháp các nhạc sỹ thay đổi cao độ của nhạc cụ để tìm ra các hòa âm tốt hơn . Gần đây cũng có một số phương pháp tối ưu được công bố như Phương pháp đàm ong nhân tạo (ABC) dựa trên phương pháp tìm kiếm thức ăn của đàm ong mật và Phương pháp tối ưu tìm kiếm nhóm (GSO) dựa trên hành vi tìm kiếm thức ăn của những con thú .

Lấy ý tưởng từ hành vi tìm kiếm thức ăn của những con kiến, phương pháp tối ưu đàm kién (ACO) cũng được công bố sau đó mang lại nhiều hứa hẹn là một thuật toán tối ưu có độ chính xác cao hơn với thời gian tính toán tốt hơn. Thuật toán được thực hiện dựa trên hành vi tìm kiếm thức ăn của đàm kién. Trong tự nhiên, đàm kién tìm kiếm thức ăn bằng cách phát hiện mùi thức ăn và theo dõi nó đến nguồn thức ăn. Các kiến đánh dấu đường đi bằng cách phân bố mùi trên đường đi của mình, người sau đó sẽ cảm thấy và đánh dấu lại đường đi này nếu thấy có mùi đường đi.

Bài báo cáo này trình bày lại thuật toán tối ưu bằng đàm kién và ý tưởng áp dụng thuật toán vào bài toán khai thác dữ liệu đảm bảo tính riêng tư, cụ thể hơn là bài toán ẩn các tập phổ biến.

2. TỐI ƯU HÓA ĐÀM KIÉN

Tối ưu đàm kién (*Ant Colony Optimization - ACO*) là một phương pháp meta-heuristic được đề xuất bởi Dorigo vào năm 1991 dựa trên ý tưởng mô phỏng cách 1920 tìm đường đi từ tổ tới nguồn thức ăn và ngược lại của các con kiến tự nhiên để giải gần đúng bài toán TUTH NP-hard.

Trên đường đi của mình các con kiến thực để lại một vết hóa chất được gọi là vết mùi (*pheromone trail*), đặc điểm sinh hóa học của vết mùi này là có khả năng út định, bay hơi và là phương tiện giao tiếp báo cho các con kiến khác thông tin về đường đi đó một cách gián tiếp. Các con kiến sẽ lựa chọn đường đi nào tồn đọng lượng mùi hay có cường độ vết mùi lớn nhất tại thời điểm lựa chọn để đi, nhờ cách giao tiếp mang tính gián tiếp và cộng đồng này mà đàm kién trong tự nhiên tìm được đường đi ngắn nhất trong quá trình tìm thức ăn mang về tổ và ngược lại. Sử dụng mô hình kiến nhân tạo này Dorigo đã xây dựng thuật toán *hệ kiến* (AS) giải bài toán người chào hàng. Thuật toán này đã được chứng minh tính hiệu quả thông qua thực nghiệm so với các mô phỏng tự nhiên khác như SA và GA. Thuật toán này về sau được phát triển và có nhiều áp dụng phong phú trong thực tế như ACS, MMAS, SMMAS,... được gọi chung là phương pháp ACO.

Theo ý tưởng này, các thuật toán ACO sử dụng thông tin heuristic kết hợp thông tin học tăng cường qua các vết mùi của các con kiến nhân tạo để giải các bài toán tối ưu tổ hợp không bằng cách đưa về bài toán tìm đường đi tối ưu trên đồ thị cấu trúc tương ứng được xây dựng từ đặc điểm của từng bài toán cụ thể. Thuật toán ACO đầu tiên là hệ kiến (*Ant System - AS*) giải bài toán Người chào hàng TSP, đến nay các thuật toán ACO đã áp dụng một cách phong phú để giải nhiều bài toán tối ưu tổ hợp khác nhau và hiệu quả nổi trội của nó đã được chứng tỏ bằng thực nghiệm.

Mô tả thuật toán ACO:

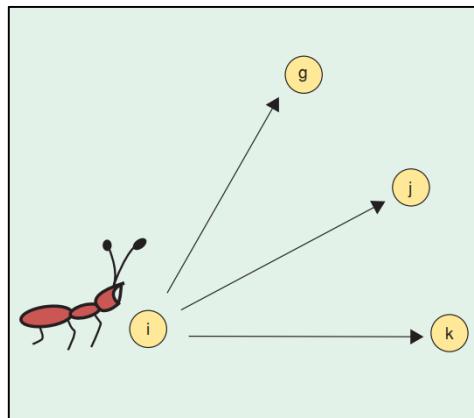
Sử dụng điều kiện kết thúc (có thể theo số bước lặp hoặc/và giới hạn thời gian chạy), ta dùng đàm kién có m con, tiến hành lặp quá trình xây dựng lời giải trên đồ thị cấu trúc $G = (V, E, H, r)$ như sau: Tại mỗi lần lặp, kiến chọn ngẫu nhiên một đỉnh $U_0 \in E_0$ làm thành phần khởi tạo $x_0 = \{u_0\}$ và thực hiện xây dựng lời giải theo thủ tục bước ngẫu nhiên. Dựa trên lời giải tìm được, đàm kién sẽ thực hiện cập nhật mùi theo cách học tăng cường.

Thủ tục bước ngẫu nhiên:

Giả sử $x_k = \langle u_0, \dots, u_k \rangle$ là mở rộng được và chưa thuộc X^* . Từ tập ràng buộc Ω , xác định tập con $J(x_k)$ của C , sao cho với $u_{k+1} \in J(x_k)$ thì $X_{k+1} = \langle u_0, \dots, u_k, u_{k+1} \rangle$ là mở rộng được. Đánh $j = u_{k+1}$ để mở rộng, được chọn với xác suất $P(j)$ như sau:

$$P(j) = \left\{ \sum \frac{[T_{ij}]^a [n_{ij}]^b}{\sum I \in J(x_k) [T_{ij}]^a [n_{ij}]^b} \right\} \in J(x_k), \quad j \in -j_{(k)}$$

Quá trình mở rộng tiếp tục cho tới khi kiếm r tìm được lời giải chấp nhận được x^r trong X^* và do đó $s^r = \varphi(x^r) \in S$.



Hình 1. Lựa chọn bước đi tiếp theo cho 2 cá thể kiến

Ta coi x^r và s^r như nhau và không phân biệt X^* với S .

Cập nhật mùi: Tùy theo chất lượng của lời giải tìm được, vết mùi trên mỗi cạnh sẽ được điều chỉnh tăng hoặc giảm tùy theo đánh giá mức độ ưu tiên tìm kiếm về sau. Lượng mùi cập nhật theo các quy tắc cập nhật mùi khác nhau sẽ cho các thuật toán khác nhau.

Vì vậy, quy tắc cập nhật mùi thường dùng làm tên gọi thuật toán, và chúng có dạng tổng quát là:

$$T_{i,j} \leftarrow (1 - p)T_{i,j} + \Delta(i,j)$$

Trong đó, p là hằng số thuộc khoảng $(0,1)$ là tỷ lệ lượng mùi bị bay hơi.

Đặc tả về thuật toán ACO:

Đầu vào:

- Khoảng cách: một ma trận chứa khoảng cách giữa mỗi thành phố
- n: số lượng kiến trong đàn
- maxIter: số lần lặp tối đa
- alpha: tham số só mủ của pheromone
- beta: tham số só mủ thông tin heuristic
- rho: tốc độ bay hơi vết pheromone
- tau0: giá trị đầu vết pheromone ban đầu
- Q: một hằng số được sử dụng để cập nhật đường dẫn pheromone

Khởi tạo:

1. Đặt đường pheromone ban đầu $\tau(i,j) = \tau_0$ cho tất cả các cạnh (i,j)
2. $t = 0$

Lặp lại cho đến khi đáp ứng tiêu chí dừng ($t < \text{maxIter}$):

1. Với mỗi con kiến k trong $\{1, \dots, n\}$:

Một. Chọn một thành phố ngẫu nhiên để bắt đầu từ

b. Lặp lại cho mỗi thành phố mà kiến k đã ghé thăm (ngoại trừ thành phố cuối cùng):

- i. Tính xác suất chuyển đến thành phố j từ thành phố i:

$p(i,j) = [\tau(i,j)^{\alpha} * h(i,j)^{\beta}] / \sum [\tau(i,l)^{\alpha} * h(i,l)^{\beta}]$ cho tất cả các phép tính
trong đó $h(i,j) = 1/\text{khoảng cách}(i,j)$ là thông tin khám phá

- ii. Sử dụng lựu chọn bánh xe roulette để chọn thành phố tiếp theo để ghé thăm dựa trên xác suất

iii. Cập nhật giải pháp hiện tại và ghi lại tổng quãng đường đã đi

- c. Cập nhật dấu vết pheromone trên mỗi cạnh trong hành trình của đàn kiến dựa trên chất lượng của giải pháp:

$$\Delta\tau(i,j) = Q / \text{tổng Khoảng cách}$$

$$\tau(i,j) = (1 - \rho) * \tau(i,j) + \rho * \Delta\tau(i,j)$$

2. Tìm k có nghiệm đúng nhất

3. Cập nhật giải pháp tốt nhất toàn cầu nếu giải pháp của kiến k tốt hơn

4. Làm bay hơi vết pheromone trên mỗi cạnh:

$$\tau(i,j) = (1 - \rho) * \tau(i,j)$$

5. Tăng t

Trả lại giải pháp tốt nhất toàn cầu

Kết thúc

Nhận xét chung về các thuật toán ACO:

Nhờ kết hợp thông tin heuristic, thông tin học tăng cường và mô phỏng hoạt động của đàn kiến, các thuật toán ACO có các ưu điểm sau:

- Việc tìm kiếm ngẫu nhiên dựa trên các thông tin heuristic trở nên linh hoạt và mềm trên miền rộng hơn so với các phương pháp heuristic đã có. Do đó, cho ta lời giải tốt hơn và có thể tìm được lời giải tối ưu.
- Học tăng cường thông qua thông tin về cường độ vết mùi cho phép từng bước thu hẹp không gian tìm kiếm, mà vẫn không loại bỏ các lời giải tốt, do đó nâng cao chất lượng thuật toán.

Chú ý: Khi áp dụng phương pháp ACO cho các bài toán cụ thể, ba yếu tố sau có ảnh hưởng quyết định đến hiệu quả thuật toán:

– **Xây dựng đồ thị cấu trúc thích hợp:** Việc xây dựng đồ thị cấu trúc để tìm được lời giải cho bài toán theo thủ tục tuần tự không khó. Khó khăn chính là với các bài toán cỡ lớn, không gian tìm kiếm quá rộng, đòi hỏi ta sử dụng các ràng buộc một cách hợp lý để giảm miền tìm kiếm của kién.

– **Chọn thông tin heuristic:** Thông tin heuristic tốt sẽ tăng hiệu quả thuật toán. Tuy nhiên, trong nhiều bài toán không có thông tin này thì có thể đánh giá chúng như nhau. Khi đó, ban đầu thuật toán chỉ đơn thuần chạy theo phương thức tìm kiếm ngẫu nhiên, vết mùi thể hiện định hướng của học tăng cường và thuật toán vẫn thực hiện được.

– **Chọn quy tắc cập nhật mùi:** Quy tắc cập nhật mùi thể hiện chiến lược học của thuật toán. Trong khi đồ thị cấu trúc và thông tin heuristic phụ thuộc vào bài toán cụ thể, quy tắc cập nhật mùi lại là yếu tố phổ dụng và thường dùng để đặt tên cho thuật toán.

3. VÍ DỤ MINH HỌA

Lấy ví dụ hàm tối ưu được định nghĩa bởi công thức:

$$f(x, y, z) = \frac{1}{2}x^2 + \frac{1}{3}y^2 + \frac{1}{6}z^2$$

Tính min value của x, y, z trong khoảng [-2,2]

Bước đầu tiên, khởi tạo quần thể các con kién. Ở đây rõ ràng bài toán có 3 biến, chọn số lượng con kién là 10. Khởi tạo, tính độ fitness ta được *Bảng 1* bên dưới.

Bảng 1. Tập hợp các con kién từ sau bước khởi tạo

	x	y	z	fitness
1	-0.231413212	-0.22266787	-0.614255563	0.105526165
2	1.139154879	0.201061281	-0.445325805	0.485823278
3	-1.261167961	0.377311053	-0.154967346	0.605365837
4	-0.281534362	-1.116092634	-0.311228113	0.665395739
5	-0.116010948	-1.380903338	-0.478561889	0.996103441
6	1.403381194	-0.74193663	-0.941943339	1.079604116
7	1.82678016	0.530788008	0.464887455	1.289263264
8	1.554425488	-0.743275028	-1.312591683	1.368791237
9	-1.726675206	0.299406381	-1.54145776	1.434639851
10	-1.556846251	1.251562107	1.213090431	1.836392002

Sau vòng lặp đầu tiên, các cá thể kién đi qua tất cả các đỉnh và tìm ra được vị trí của thức ăn tốt nhất ở vòng lặp đầu tiên là [0.3161076902875086, 0.2797437968345613, 0.30768903059761277, 0.08821507647835909]

0.3348947825492	0.1251796619854	-0.2127888859948	0.05276633068079
-----------------	-----------------	------------------	------------------

Như vậy ta đã thu được vị trí thức ăn tốt nhất, tương tự thực hiện lặp lại 10 vòng rồi đến 50 vòng, 100 vòng, ...

Bảng 2. Tập hợp các con kién từ sau 10 vòng lặp

1	0.31610769	0.279743797	0.307689031	0.088215076
2	0.31610769	0.124798633	-0.080581778	0.018616403

3	0.170998387	0.124798633	-0.080581778	0.018616403
4	0.170998387	0.124798633	-0.080581778	0.018616403
5	0.170998387	0.124798633	-0.080581778	-0.080581778
6	0.007604985	0.184772535	-0.001690199	0.017090199
7	0.007604985	0.184772535	-0.001690199	0.017090199
8	0.007604985	0.184772535	-0.001690199	0.017090199
9	0.007604985	0.184772535	-0.001690199	0.017090199
10	0.007604985	0.184772535	-0.001690199	0.017090199

Bảng 3. Tập hợp các con kiến từ sau 100 vòng lặp

91	0.051920782	-0.033362629	0.090560405	0.002821986
92	0.051920782	-0.033362629	0.090560405	0.002821986
93	0.051920782	-0.033362629	0.090560405	0.002821986
94	0.051920782	-0.033362629	0.090560405	0.002821986
95	0.051920782	-0.033362629	0.090560405	0.002821986
96	0.051920782	-0.033362629	0.090560405	0.002821986
97	0.051920782	-0.033362629	0.090560405	0.002821986
98	0.051920782	-0.033362629	0.090560405	0.002821986
99	0.051920782	-0.033362629	0.090560405	0.002821986
100	0.051920782	-0.033362629	0.090560405	0.002821986

4. ÚNG DỤNG TRONG BÀI TOÁN ẨN TẬP PHỔ BIẾN

Định nghĩa 1. (Bài toán ẩn tập phổ biến). Cho trước một cơ sở dữ liệu giao tác D có tập hạng mục $I = \{i_1, i_2, \dots, i_r\}$, một độ hỗ trợ tối thiểu δ và một tập các tập phổ biến nhạy cảm SIs. Thực hiện biến đổi D thành D' sao cho $\text{supD}'(X) < \delta \forall X \in S$. Khi thực hiện bài toán ẩn tập phổ biến, thông thường xuất hiện 3 loại kết quả không mong muốn như sau.

Định nghĩa 2. (Fail to Hide – FTH). Là số lượng các tập nhạy cảm xuất hiện trong cơ sở dữ liệu D' . Điều này có nghĩa là sao khi thực hiện phương pháp ẩn tập nhạy cảm, có một số tập không thể ẩn được thông qua phương pháp đó. Thông số này được ký hiệu là α .

Định nghĩa 3. (Not to be Hidden - NTH). Là số lượng các tập phổ biến không nhạy cảm bị ẩn trong cơ sở dữ liệu D' . Điều này có nghĩa là khi áp dụng phương pháp ẩn tập nhạy cảm, một số tập phổ biến không nhạy cảm cũng bị ẩn theo. Thông số này ký hiệu là β .

Định nghĩa 4. (Not to be Generated - NTG). Là số lượng các tập không phổ biến trong cơ sở dữ liệu D lại trở thành phổ biến trong cơ sở dữ liệu D' . Điều này có nghĩa là khi áp dụng phương pháp ẩn tập nhạy cảm, một số tập không phổ biến trong cơ sở dữ liệu ban đầu lại trở thành phổ biến trong cơ sở dữ liệu hiệu chỉnh, các tập này còn gọi là các tập ma. Thông số này được ký hiệu là γ .

Thực tế, không có phương pháp ẩn tập phổ biến nào hoàn hảo đến mức có thể làm cho các thông số không mong muốn tất cả đều bằng không. Như vậy, bài toán ẩn tập phổ biến có thể quy về bài toán tối ưu với hàm mục tiêu

$$f = w_1\alpha + w_2\beta + w_3\gamma$$

Trong đó, w1, w2, w3 là các trọng số tương ứng với các thông số không mong muốn. Thông thường w1 sẽ được đặt lớn hơn nhiều so với w2 và w3 với mong muốn có thể ăn hoàn toàn các tập nhạy cảm.

Áp dụng phương pháp tối ưu đòn kién cho bài toán ăn tập nhạy cảm, công trình này đề xuất một phương pháp mới để giải bài toán theo hướng metaheuristic. Mỗi đòn kién sẽ chứa N_var con kién. Mỗi con kién sẽ chứa thông tin về Tid (Transaction ID) cần hiệu chỉnh hoặc xóa bỏ. N_var có thể tính theo công thức

$$N_{var} = \left\lceil \frac{Max_sup(si) - \delta * |D|}{1 - \delta} \right\rceil$$

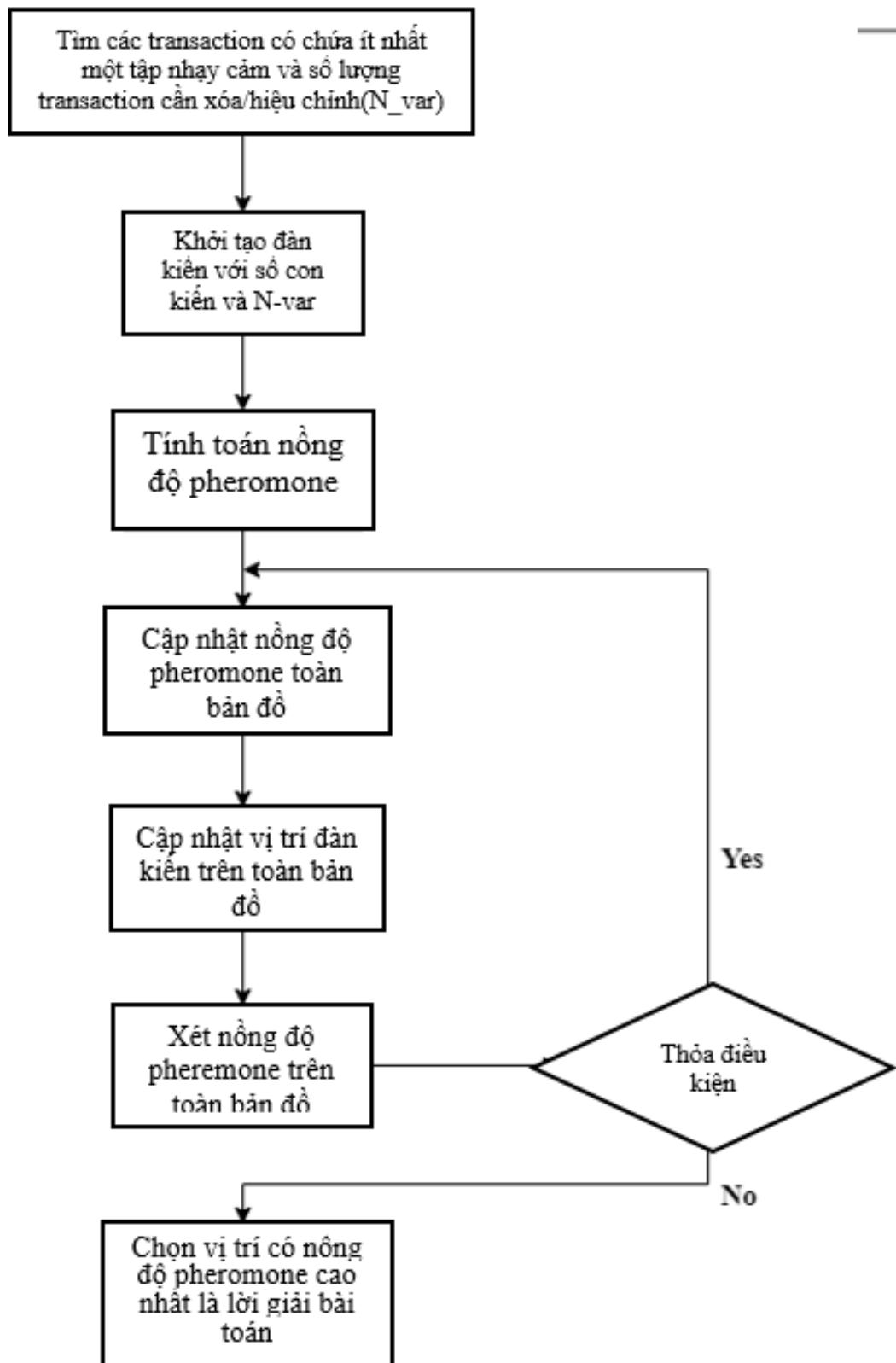
Trong đó:

- Max_sup(si): độ hỗ trợ lớn nhất trong số tất cả các tập nhạy cảm δ: độ hỗ trợ tối thiểu.
- |D|: số lượng các giao tác có trong tập dữ liệu gốc.

Sau khi tính được nồng độ Pheromone tối ưu, các con kién sẽ được khởi tạo một cách ngẫu nhiên dựa trên các thông tin các transaction có chứa ít nhất 1 tập nhạy cảm. Các con kién đã được khởi tạo sẽ được đánh giá độ thích nghi sau đó và sắp xếp lại theo độ thích nghi.

Trong mỗi vòng lặp, nồng độ pheromone sẽ được cập nhật dựa trên quá trình đi tìm kiếm thức ăn của đòn kién. Đòn kién sẽ di chuyển về phía thức ăn dựa vào nồng độ pheromone đã được cập nhật.

Các con kién sẽ đánh giá nồng độ pheromone và chèn vào trong danh sách vị trí ứng với nồng độ pheromone của nó. Vị trí có nồng độ pheromone thấp nhất sẽ bị loại bỏ. Phương pháp ăn tập phổ biến sử dụng tối ưu đòn kién được biểu thị bởi sơ đồ ở Hình 2.



Hình 2. Sơ đồ phương pháp giải bài toán ẩn tập phô biến sử dụng ACO

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này trình bày thuật toán tối ưu hóa đàm kién bằng cách tạo ra các con kiến đi tìm kiếm thức ăn. Các con kiến đi tìm kiếm thức ăn, mô phỏng bằng các đối tượng ảo được gọi là "điểm kiểm tra". Thuật toán sử dụng các "mùi hương" để hướng dẫn các con kiến ảo đi theo hướng trong quá trình tìm kiếm. Các mùi hương này đại diện cho các đường đi được tìm thấy bởi các con kiến ảo khác và có độ hấp dẫn khác nhau tùy thuộc vào chi phí của đường đi. Các con kiến ảo sẽ ưu tiên đi theo những đường có mùi hương hấp dẫn hơn, và các mùi hương này càng mạnh hơn nếu các con kiến ảo trên đường đó tìm thấy lời giải tốt hơn. Trong bài báo cũng trình bày phương pháp để xuất ứng dụng thuật toán tối ưu đàm kién để giải bài toán khai thác dữ liệu đảm bảo tính riêng tư. Mặc dù chưa có thực nghiệm cụ thể về phương pháp mới này, nhưng phương pháp hứa hẹn sẽ cho kết quả chính xác.

Các kết quả thực nghiệm cụ thể sẽ được công bố trong các báo cáo tiếp theo.

TÀI LIỆU THAM KHẢO

1. "Ant Colony Optimization: A Review," by Dorigo, M., Blum, C. (2005) in Artificial Life Journal.
2. "Ant Colony Optimization for the Traveling Salesman Problem: An Implementation Tutorial," by Gambardella, L.M., Dorigo, M. (1996) in Journal of Artificial Intelligence Research.
3. "Ant Colony Optimization: A Metaheuristic Approach to Combinatorial Optimization Problems," by Stützle, T. (1998) in Theoretical Computer Science..
4. "A survey of ant colony optimization algorithms for the minimum spanning tree problem," by Wang, Y., Liu, L., & Gao, H. (2013) in Journal of Computational and Applied Mathematics.
5. "An Ant Colony Optimization Algorithm for Job Shop Scheduling," by Dorigo, M., Di Caro, G. (1999) in IEEE Transactions on Evolutionary Computation

ABSTRACT

OPTIMIZATION OF ANSWERS AND APPLICATIONS IN HIDING FREQUENT ITEMSET PROBLEMS

Ho Thanh Hai¹, Dinh Nguyen Trong Nghia¹

¹*Ho Chi Minh City University of Food Industry*

*Email: nghiadnt@hufi.edu.vn

This paper presents the ant colony optimization algorithm which is a metaheuristic algorithm inspired by the foraging behavior of ants and borers. The strength of the ant colony optimization algorithm is its ability to find optimal solutions in a generalization problem of large size. In addition, the paper also proposes a method of applying ant colony optimization to the problem of hiding frequent itemsets. Because of the advantages of ant colony optimization, it is possible to find the optimal solution for the problem of hiding frequent itemsets, minimizing side effects.

Keywords: ant colony optimization, ACO, PPDM, hiding frequent itemset.