

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, Đinh 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, Đinh Nguyễn Trọng Nghĩa^{1,*}* **288**
27. Tối ưu hóa sói xám và ứng dụng trong bài toán ẩn tập phô biến - *Phạm Nguyễn^{1,*}, Đinh 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 TÌM KIẾM CHIM HỒNG HẠC VÀ ỨNG DỤNG TRONG BÀI TOÁN ẨN TẬP PHỔ BIẾN

Lê Huỳnh Đức¹, Nguyễn Thị Hằng¹, 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 giới thiệu thuật toán tối ưu hóa bầy đàn thông minh, tên là "Tìm kiếm chim hồng hạc" (FSA), dựa trên hành vi di cư và tìm kiếm thức ăn của chim hồng hạc. Thuật toán này có khả năng tìm kiếm giải pháp tối ưu cho các bài toán tối ưu hóa nhanh chóng và hiệu quả. FSA sử dụng các thông số như số lần lặp tối đa, quy mô quần thể chim hồng hạc, giới hạn khoảng tìm kiếm, số lượng đàn chim hồng hạc di cư và kiếm ăn. Bài báo cũng đề xuất áp dụng FSA vào bài toán ẩn tập phổ biến, giúp tìm giải pháp tối ưu và giảm thiểu rủi ro liên quan đến cạnh tranh giữa các cá thể. Thuật toán FSA phân bố các cá thể đều trên không gian tìm kiếm và di chuyển đến các vị trí có nhiều thông tin hữu ích, giúp tìm kiếm giải pháp tốt nhất.

Từ khóa: Flamingo Search Algorithm, FSA, PPDM, hiding frequent itemset

1. GIỚI THIỆU

Flamingo Search Algorithm (FSA) là một thuật toán tối ưu hóa được lấy cảm hứng từ hiện tượng chim hồng hạc di cư. Ý tưởng của thuật toán là mô phỏng lại cách mà các con chim hồng hạc tìm kiếm lấy nguồn thức ăn trên mặt nước. Các con chim hồng hạc này sử dụng mỏ của mình để đào xuống đáy nước và tìm kiếm thức ăn. Thuật toán FSA sử dụng quy trình tìm kiếm ngẫu nhiên, kết hợp với quy trình cập nhật vị trí của các cá thể để tìm kiếm các giá trị tối ưu. Flamingo Search Algorithm thực hiện tìm kiếm trong không gian bài toán bằng cách sử dụng việc di chuyển ngẫu nhiên trong không gian tìm kiếm. Thuật toán sử dụng một bộ các con số ngẫu nhiên để điều khiển việc di chuyển của các hồng hạc trong quá trình tìm kiếm. Mỗi lần di chuyển, vị trí của các hồng hạc được cập nhật dựa trên một công thức tính toán giữa vị trí hiện tại và các con số ngẫu nhiên. Quá trình này được lặp lại cho đến khi đạt được kết quả tìm kiếm mong muốn hoặc đạt đến điều kiện dừng được thiết lập trước đó. Thuật toán tối ưu hóa ngẫu nhiên có ưu điểm là: dễ triển khai, dễ thích nghi và có tốc độ ổn định, độ bền và khả năng mở rộng, ít giám sát hơn [1].

So với các thuật toán khác thì thuật toán tối ưu hóa ngẫu nhiên có tính năng cao hơn. Một loại thuật toán tối ưu hóa ngẫu nhiên điển hình là thuật toán trí thông minh bầy đàn tìm ra đường dẫn hội tụ và giải pháp tối ưu một cách độc đáo. Các thuật toán tối ưu hóa trí thông minh bầy đàn sử dụng ít tài nguyên, có khả năng tìm kiếm toàn cục, dễ dàng triển khai, có khả năng tìm kiếm đa nhiệm. Trong những năm gần đây, nhiều thuật toán tối ưu hóa trí thông minh bầy đàn mới đã được đề xuất, chẳng hạn như Whale Optimization Algorithm (WOA) [2], Tunicate Swarm Algorithm (TSA) [3], Sparrow Search Algorithm (SSA) [4], Particle Swarm Optimization (PSO). Các thuật toán này cung cấp cho kết quả tìm kiếm nhanh nhưng cũng còn hạn chế.

Bên cạnh đó tất cả thuật toán vừa nêu trên đều có ưu điểm chung là khả năng tìm kiếm tối ưu toàn cục, tốc độ tìm kiếm nhanh, khả năng tìm kiếm đa nhiệm, dễ triển khai. Chúng được sử dụng rộng rãi trong việc giải quyết các vấn đề tối ưu hóa hàm mục tiêu, tối ưu hóa mô hình máy học, tối ưu hóa các thiết kế kỹ thuật, tối ưu hóa các quyết định kinh doanh và tài chính, tối ưu hóa quy trình sản xuất, v.v...

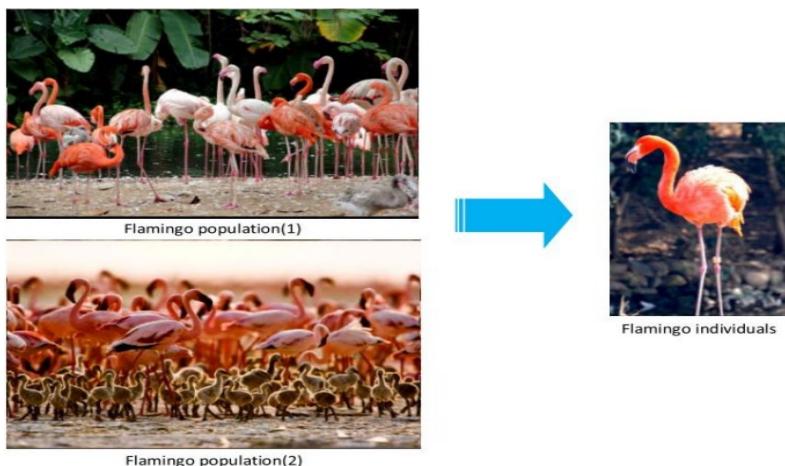
Thuật toán WOA lấy cảm hứng từ hành vi săn mồi của cá voi, TSA lấy cảm hứng từ cách di chuyển của loài động vật biển lục tunicate, thuật toán SSA lấy cảm hứng từ việc tìm kiếm thức ăn của chim sẻ, PSO lấy cảm hứng từ cách đàn chim tìm kiếm thức ăn trong tự nhiên. Do sự phát triển của công nghệ kỹ thuật và các vấn đề kỹ thuật ngày càng phức tạp dẫn đến mong muốn mở rộng tính đa dạng của thuật toán trí thông minh bầy đàn và cải thiện hiệu suất. Để hiệu suất thuật toán ngày càng tối ưu thì phải tăng khả năng khám phá, phát triển không gian tìm kiếm. Nghĩa là, hiệu suất của thuật toán có thể giảm sau khi giải các bài toán khác có tính chất khác.

Bài báo cáo này trình bày lại thuật toán tối ưu Flamingo 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 phô biến.

2. THUẬT TOÁN TÌM KIẾM CHIM HỒNG HẠC

2.1 Đặc điểm của thuật toán

Chim hồng hạc là loài chim di cư sống theo bầy đàn, thức ăn chủ yếu là tảo, tôm nhỏ, nghêu, giun nhỏ và ấu trùng côn trùng. Chim hồng hạc cũng kiếm ăn theo một cách đặc biệt, bằng cách cúi chiếc cổ dài xuống và quay đầu lại, sau đó đi bộ. Trong khi quét chiếc mỏ cong của chúng quanh cơ thể và chạm vào đáy nước để kiếm ăn [5]. Các quần thể và cá thể chim hồng hạc trong tự nhiên được thể hiện trong Hình 1.



Hình 1. Quần thể và cá thể chim hồng hạc trong tự nhiên

Hai đặc điểm tập tính chính của chim hồng hạc là tập tính kiếm ăn và di cư. Quần thể chim hồng hạc chủ yếu sinh sống ở những khu vực có nhiều thức ăn. Sau một thời gian kiếm ăn rộng rãi, quần thể chim hồng hạc di cư khi thức ăn trong khu vực bị giảm đến mức không

thể đáp ứng quần thể. Bài báo này thiết lập mô hình tìm kiếm thức ăn và mô hình di cư tương ứng.

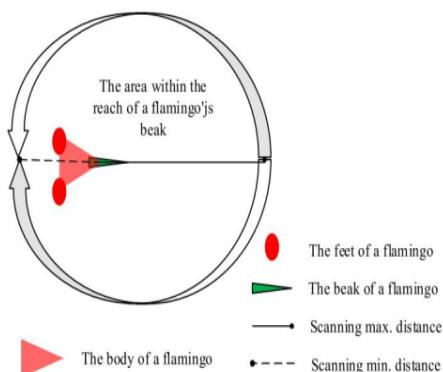
Các ý tưởng tối ưu hóa chính của mô hình FSA được trình bày như sau:

Chim hồng hạc hát cho nhau nghe để thông báo vị trí của chúng và thức ăn tại địa điểm. Quần thể chim hồng hạc không biết nơi nào có nhiều thức ăn nhất trong khu vực tìm kiếm hiện tại. Thay vào đó, chúng cập nhật qua tiếng hát của đồng loại xung quanh. Hành vi này của hồng hạc phù hợp với ý tưởng tối ưu hóa trí thông minh bầy đàn. Và chúng ta không biết không gian tìm kiếm là bao nhiêu, tác nhân tìm kiếm là chim hồng hạc, hồng hạc khám phá ra không gian tìm kiếm và trao đổi thông tin với nhau và cố định vị trí tìm thấy đó. Từ đó rút ra được giải pháp tối ưu. Quy tắc thay đổi vị trí dựa trên hành vi của chim hồng hạc. Hành vi chính của nó chủ yếu có hai loại: hành vi kiếm ăn và hành vi di cư. Hành vi kiếm ăn có thể được chia thành hai đặc điểm hành vi: Quét mỏ và di chuyển chân

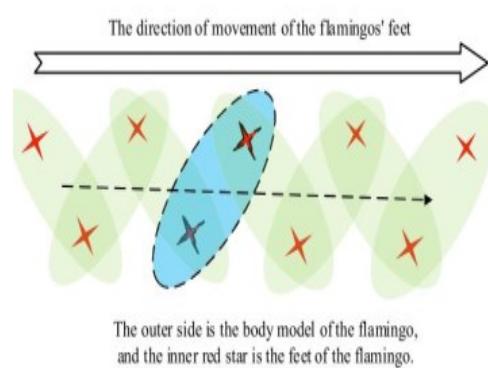
2.2 Ý tưởng và mô hình toán học của thuật toán

Tìm hiểu hành vi: Hành vi giao tiếp khi con hạc phát hiện chỗ có thức ăn nó sẽ gọi những con hạc khác đến vị trí đó. Chỗ tụ tập nhiều nhất là chỗ có nhiều thức ăn nhất. Theo lý thuyết những con hạc này không biết chỗ nào có nhiều thức ăn. Nhưng trong thuật toán chúng ta có thể biết vì ta có thể thiết lập điều kiện kết thúc chương trình.

FSA là một thuật toán mô phỏng chim hồng hạc có gắng tìm giải pháp tối ưu trong khu vực tìm kiếm (tức là vị trí có nhiều thức ăn nhất) dựa trên nguồn thông tin hạn chế có sẵn.



Hình 2. Hành vi quét mỏ



Hình 3. Hành vi di chuyển hai chân

Mỏ của chim hồng hạc khi úp ngược trong nước có chức năng tương tự như một cái sàng lớn, giúp hút nước và lọc thức ăn nhanh chóng nhờ vào các rãnh sâu ở mỏ dưới và các rãnh nông, có nắp ở mỏ trên. Khi chim hồng hạc đang kiếm ăn, chúng cúi đầu, lộn miếng ngược và ăn thức ăn vào miệng, sau đó thải ra nước thừa và cặn bã không ăn được. Cách kiếm ăn này phụ thuộc vào sự dồi dào thức ăn trong khu vực. Khi mỏ của chim hồng hạc quét qua một khu vực có nhiều thức ăn hơn, chúng sẽ quét khu vực đó một cách cẩn thận hơn và cố của chim sẽ dần duỗi ra, làm tăng bán kính quét của mỏ. Xác suất quét khu vực để tìm thức ăn cũng sẽ tăng lên. Mô hình hành vi quét mỏ của chim hồng hạc được mô tả trong Hình 2.

Khi có nhiều con chim hồng hạc trong quần thể đồng thời hội tụ tại một vị trí, khả năng có nhiều thức ăn ở khu vực đó càng cao. Bài báo này mô phỏng hành vi quét mỏ của chim hồng hạc trong không gian đa chiều của quần thể được ký hiệu là x_{ij} , và mô hình tính đến sự

thay đổi trong lựa chọn của từng cá thể chim hồng hạc trong tự nhiên cũng như tác động của sự biến động đột ngột của môi trường đến hành vi kiếm ăn của chúng. Hành vi tìm kiếm thức ăn của từng con chim hồng hạc sẽ có sai số khi truyền thông tin. Để mô phỏng sai số này, ta sử dụng một phân phối ngẫu nhiên theo phân phối chuẩn, trong đó việc quét mỏ của chim hồng hạc có xác suất cao hướng về vị trí có nhiều thức ăn nhất. Tuy nhiên, cũng có một xác suất nhỏ cho sai sót dựa trên thông tin này.

Tiếp theo, Để có thể đo lường khoảng cách tối đa mà mỏ chim hồng hạc quét qua trong quá trình tìm kiếm thức ăn được biểu diễn bằng công thức: $|G_1 \times x_{ij} + \varepsilon_2 \times x_{ij}|$ trong đó ε_2 là một số ngẫu nhiên có giá trị -1 hoặc 1. Mục đích chính của khoảng cách tối đa này là mở rộng phạm vi tìm kiếm của mỏ chim hồng hạc trong quá trình tìm thức ăn. Trong đó, G_1 là một số ngẫu nhiên tuân theo phân phối chuẩn thông thường trong khoảng $N(0,1)$. Để mô phỏng phạm vi quét mỏ của chim hồng hạc trong hành vi tìm kiếm có thể được nói là quét mỏ lần hai, và cũng giống công thức quét mỏ lần một thì lại thêm G_2 được biểu diễn bằng công thức: $G_2 \times |G_1 \times xb_j + \varepsilon_2 \times x_{ij}|$ trong đó G_2 là một số ngẫu nhiên tuân theo phân phối chuẩn thông thường trong khoảng $N(0,1)$.

Mô hình hành vi di chuyển bằng hai chân của hồng hạc được thể hiện trong Hình 3. Khi hồng hạc kiếm thức ăn, trong lúc quét bằng mỏ, móng vuốt của chúng sẽ di chuyển về phía nơi có nhiều thức ăn nhất. Nói một cách dễ hiểu là chúng vừa di chuyển vừa quét mỏ. Giả sử vị trí nơi thức ăn phong phú nhất trong quần thể là xb_j , khoảng cách đã đi được đo bằng " $\varepsilon_1 \times xb_j$ ", trong đó " ε_1 " là một số ngẫu nhiên -1 hoặc 1, chủ yếu là để tăng phạm vi tìm kiếm của hồng hạc và định lượng sự khác biệt của cá thể trong việc lựa chọn.

Tổng kết lại, bước di chuyển của chim hồng hạc trong quá trình tìm kiếm thức ăn ở vòng lặp thứ t là tổng khoảng quét mỏ chim hồng hạc cộng với khoảng cách di chuyển của hai chân, như được hiển thị trong công thức số (1).

$$b_{ij}^t = \varepsilon_1 \times xb_j^t + G_2 \times |G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t| \quad (1)$$

Công thức cập nhật vị trí kiếm ăn của chim hồng hạc được thể hiện như sau:

$$x_{ij}^{t+1} = (x_{ij}^t + \varepsilon_1 \times xb_j^t + G_2 \times |G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t|)/K \quad (2)$$

Trong công thức (2), x_{ij}^{t+1} đại diện vị trí của con hồng hạc thứ i trong chiều thứ j của quần thể trong lần lặp thứ $t+1$. x_{ij}^t đại diện vị trí của con hồng hạc thứ i trong chiều thứ j trong lần lặp thứ t của quần thể chim hồng hạc cụ thể là vị trí bàn chân chim hồng hạc xb_j^t : đại diện cho vị trí chiều thứ j của chim hồng hạc có độ thích nghi tốt nhất trong quần thể ở lần di cư. $K = K(n)$: là hệ số khuếch tán, là một số ngẫu nhiên tuân theo phân phối khi bình phương n bậc tự do. Nó được sử dụng để tăng quy mô phạm vi kiếm ăn của chim hồng hạc và để mô phỏng cơ hội lựa chọn cá thể trong tự nhiên, tăng khả năng tìm kiếm toàn cầu của nó $G_1 = G_2 = N(0,1)$ là một số ngẫu nhiên tuân theo phân phối chuẩn thông thường. ε_1 và ε_2 là một số ngẫu nhiên -1 hoặc 1.

Khi thức ăn ít trong khu vực tìm kiếm hiện tại, quần thể hồng hạc di cư đến khu vực tiếp theo có thức ăn phong phú hơn. Giả sử vị trí của khu vực giàu thức ăn trong chiều thứ j là xb_j , công thức cho sự di cư của quần thể hồng hạc như sau:

$$x_{ij}^{t+1} = x_{ij}^t + \omega \times (xb_j^t - x_{ij}^t) \quad (3)$$

Trong công thức (3), x_{ij}^{t+1} đại diện vị trí của con hồng hạc thứ i trong chiều thứ j của quần thể trong lần lặp thứ t+1. x_{ij}^t đại diện vị trí của con hồng hạc thứ i trong chiều thứ j trong lần lặp thứ t của quần thể chim hồng hạc cụ thể là vị trí bàn chân chim hồng hạc xb_j^t : đại diện cho vị trí chiều thứ j của chim hồng hạc có độ thích nghi tốt nhất trong quần thể ở lần di cư. $\omega = N(0, n)$: là một số ngẫu nhiên Gaussian với n bậc tự do, được sử dụng để tăng không gian tìm kiếm trong quá trình di cư của chim hồng hạc và mô phỏng tính ngẫu nhiên của các hành vi riêng lẻ của chim hồng hạc trong quá trình di cư cụ thể.

Mô tả thuật toán tìm kiếm chim hồng hạc bao gồm:

- Di chuyển và quét mỏ trong khu vực tìm kiếm.
 - Di cư lần 1 một số cá thể ở đầu để mở rộng phạm vi tìm kiếm, Xong tiếp tục di cư lần 2 là những cá thể chim hồng hạc còn lại.
 - Cập nhật vị trí của cá thể chim hồng hạc có độ thích nghi tốt nhất so với ban đầu.
- Thuật toán FSA có thể được tinh chỉnh thông qua các tham số như số lượng cá thể, tỷ lệ lai ghép, tỷ lệ đột biến và tỷ lệ hoán đổi để tối ưu hóa quá trình tìm kiếm.

2.3 Quy trình cơ bản của FSA

Bước 1: Khởi tạo quần thể, quần thể được đặt tên là P, số vòng lặp tối đa được đặt tên là IterMax và tỷ lệ chim hồng hạc di cư trong phần đầu tiên được đặt tên là MPb.

Bước 2: Số lượng chim hồng hạc tìm kiếm thức ăn trong vòng lặp thứ i của việc cập nhật quần thể chim hồng hạc là MPr = rand [0,1] × P × (1-MPb). Số lượng chim hồng hạc di cư trong phần đầu tiên của vòng lặp này là MPo = MPb × P. Số lượng chim hồng hạc di cư trong phần thứ hai của vòng lặp này là MPt = P - MPo - MPr. Giá trị thích nghi của các con chim được xác định, và quần thể chim hồng hạc được sắp xếp theo giá trị thích nghi của từng con chim. Các chim hồng hạc MPb có giá trị thích nghi thấp và các chim hồng hạc MPt có giá trị thích nghi cao được coi là chim hồng hạc di cư, trong khi các con khác được coi là chim hồng hạc tìm kiếm thức ăn.

Bước 3: Các con chim hồng hạc di cư được cập nhật theo công thức (3), và các con chim hồng hạc tìm kiếm thức ăn được cập nhật theo công thức (2).

Bước 4: Kiểm tra các con chim hồng hạc nào bị vượt ra ngoài giới hạn (xem mã giả để biết chi tiết).

Bước 5: Nếu đã đạt đến số lượng vòng lặp tối đa, chuyển đến Bước 6; nếu không, chuyển đến Bước 2.

Bước 6: Đưa ra giải pháp tối ưu và giá trị tối ưu.

Lưu ý rằng, trong quá trình thực hiện Bước 2 đến Bước 5, nếu tìm thấy giải pháp tối ưu trước khi đạt đến số lần lặp tối đa, thì thuật toán cũng sẽ dừng lại và chuyển đến Bước 6.

Thuật toán 1: FSA

Input: Hàm mục tiêu fitness ()

Quy mô quần thể chim hồng hạc Pop
 Số lần lặp tối đa MaxIter
 Số chiều của không gian tìm kiếm Dim
 Biên dưới của khoảng tìm kiếm lb
 Biên trên của khoảng tìm kiếm ub
 Sinh số ngẫu nhiên với hai giá trị -1 hoặc 1 rand_1
 Sinh số ngẫu nhiên trong khoảng 0 đến 1 Vs
 Số lượng chim hồng hạc di cư trong đợt đầu tiên PMc
 Số lượng đàn hồng hạc di cư đợt hai PMf
 Số lượng chim hồng hạc kiêm ăn Pmu
 Mảng lưu con có giá trị tốt nhất Xb

Hàng số

Tỷ lệ di cư cơ bản của chim hồng hạc trong giai đoạn đầu tiên MP_b = 0.1

Khởi tạo:

Tạo ra một mảng chứa vị trí của các con hồng hạc(positions).

for each i to Pop do

 for j = 1 dim do

 positions [i, j] = lb + (ub - lb) * rand ()

 end for

end for

// Sắp xếp các con chim hồng hạc theo fitness từ tốt nhất đến xấu nhất

Sort_pop(X,fitness)

// Lấy ra giá trị tốt nhất

GbestPositon = positions [0]

GbestScore = fitness (GbestPositon)

Tối ưu:

For i = 1 to MaxIter do

 Vs = rand ()

 PMf = int((1-Mp_b) * Vs*Pop)

 PMc = MP_b * Pop

 Pmu = Pop - PMc -PMf

 Xb = positions [0]

 For j to PMc do

 For z to dim do

```

A1 = rmgnormal (loc=0, scale=1.2, size=1)
positions [j, z] = positions [j, z] + (Xb[i] – positions [j, i]) * A1
End for
End for
For int (PMc) to int (PMc+PMu) do
    For z to dim do
        // sinh ra một số ngẫu nhiên theo phân phối chi-square (rng.chisquare(df=8, size=1)
        positions[j,i]=(positions[j,i]+rand_1() * Xb[i] + np.random.randn() * (np.random.randn()* np.abs(Xb[i]
        + rand_1() * positions[j, i]))) / (rng.chisquare(df=8, size=1))
    End for
    End for
    For int (PMc+PMu) to Pop do
        For z to dim do
            // A1 là số ngẫu nhiên theo phân phối chuẩn Gaussian
            A1 = rmgnormal (loc=0, scale=1.2, size=1)
            Positions [j, i] = positions [j, i] + (Xb[i] – positions [j, i]) * A1
        End for
        For j = 0 to Pop
            Positions[i] = clip (Positions[i], lb, ub)
        End for
        Sort (Positions, fitness)
        If (fitness (Positions [0]) <= GbestScore)
            GbestScore = fitness (Positions [0])
            GbestPositon = Positions [0]
        End if
    End for
Output: trả về vị trí tốt nhất GbestPositon và độ thích nghi GbestScore

```

Dòng lệnh được đánh dấu là “chiến lược tìm kiếm tối ưu” là thành phần quan trọng nhất của FSA và là hai biểu thức toán học quan trọng nhất trong thuật toán FSA. Trong đó biểu thức (2) là hành vi kiểm ăn và cuối cùng biểu thức (3) là để chỉ hành vi di cư đợt một và đợt hai tuy nhiên cả 2 lần di cư chỉ khác phạm vi trong thuật toán tìm kiếm chim hồng hạc.

$$x_{ij}^{t+1} = (x_{ij}^t + \varepsilon_1 \times xb_j^t + G_2 \times |G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t|) / K \quad (2)$$

$$x_{ij}^{t+1} = x_{ij}^t + \omega \times (xb_j^t - x_{ij}^t) \quad (3)$$

3. VÍ DỤ MINH HỌA

Lấy ví dụ hàm tối ưu được định nghĩa bởi

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

Trong đó x, y, z sẽ nằm trong khoảng [-2, 2].

Bước đầu tiên khởi tạo quần thể chim hòng hạc ngẫu nhiên. Rõ ràng ở đây bài toán có 3 biến x y z theo ví dụ đã cho số lượng chim hòng hạc trong quần thể là 10 con chim. Khởi tạo, tính độ thích nghi và sắp xếp theo độ thích nghi từ nhỏ nhất đến lớn Bảng 1 như sau:

Bảng 1. Tập hợp các hạt điện tử sau bước khởi tạo

Index	x	y	z	fitness
0	0.018317705	-0.954501759	-0.07420584	0.456566401
1	1.576923786	0.710334211	0.46732492	1.117582319
2	1.763710628	0.244216324	1.0800412	1.261127364
3	-1.95104432	-0.151894247	0.42247673	1.310141675
4	1.598244066	-0.496744224	1.76238485	1.492505502
5	1.462621843	1.111646049	1.51871758	1.715383203
6	-0.64193812	-1.832888837	0.86677271	1.942318085
7	1.68261582	-1.555546307	-1.57806746	2.568643639
8	-1.65311346	-1.831141998	0.13338407	2.590433766
9	1.693722415	-1.825758368	-1.21112253	2.867398314

Từ Bảng 1, có thể nhìn rõ và xác định được một cá thể đầu đàn có chỉ số thích nghi tốt nhất sẽ được chọn và tiến hành di cư cho quần thể đột một. Áp dụng thuật toán ta sẽ sinh ra cá thể mới từ lần di cư đột một theo tỉ lệ di cư $0.2 * 10 = 2$ (0.2 là tỉ lệ di cư cho trước của bài toán 10 là số lượng cá thể khởi tạo) Tuy nhiên thuật toán tự động bỏ qua cá thể tốt nhất là con đầu đàn được Bảng 2 như sau:

Bảng 2. Các cá thể sau khi di cư đột một

Index	x	y	z
0	0.01831771	-0.95450176	-0.07420584
1	2.52184203	2.25801274	1.29357125
2	1.76371063	0.24421632	1.08004119
3	-1.95104432	-0.15189425	0.42247673
4	1.59824407	-0.49674422	1.76238485
5	1.46262184	1.11164605	1.51871758
6	-0.64193812	-1.83288884	0.86677271
7	1.68261582	-1.55554631	-1.57806746
8	-1.65311346	-1.831142	0.13338407
9	1.69372242	-1.82575837	-1.21112253

Sau khi di cư đợt đầu tiếp tục cho số lượng chim di chuyển bằng hai chân và quét mỏ kiềm ăn, VS = 0.50273078529049419 (Số ngẫu nhiên theo phân phối chuẩn), di cư đợt đợt 2 PMF = 4, Di cư đợt đầu PMc = 2, (PMc) to int (PMc+PMu) Số lượng chim kiềm ăn là 4 trong phạm vi quần thể từ 2 đến 6 sẽ thu được các vị trí mới được tô đậm dưới bảng và các vị trí các con chim được hiệu chỉnh theo công thức thuật toán thu được kết quả Bảng 3 bên dưới:

Bảng 3. Tập hợp các cá thể kiềm ăn

Index	x	y	Z
0	0.01831771	-0.95450176	-0.07420584
1	2.52184203	2.25801274	1.29357125
2	-0.21671154	0.42137008	0.12696972
3	-0.25529128	0.06973586	0.04292713
4	0.07093222	-0.26234487	0.21619166
5	0.10262446	0.00257076	0.12621516
6	-0.64193812	-1.83288884	0.86677271
7	1.68261582	-1.55554631	-1.57806746
8	-1.65311346	-1.831142	0.13338407
9	1.69372242	-1.82575837	-1.21112253

Tiếp theo sau đó cuộc di cư đợt 2 sẽ được diễn ra cho các cá thể còn lại cũng áp dụng thuật toán lại thu được kết quả như Bảng 4 dưới đây:

Bảng 4. Các cá thể di cư đợt hai

Index	x	y	Z
0	0.01831771	-0.95450176	-0.07420584
1	2.52184203	2.25801274	1.29357125
2	-0.21671154	0.42137008	0.12696972
3	-0.25529128	0.06973586	0.04292713
4	0.07093222	-0.26234487	0.21619166
5	0.10262446	0.00257076	0.12621516
6	-0.15790724	-1.02718838	0.82213052
7	2.49256880	-1.42660959	-3.98628412
8	0.69234925	-2.15563961	0.64428492
9	2.44357811	-2.91530343	-2.72947764

Quá trình di cư 2 hoàn tất theo sau đó phải kiểm tra lại ranh giới như đã cho ban đầu của bài toán nếu vượt quá biên trên thì phải cho bảng lại biên dưới và ngược lại tương tự (Ví dụ: Ở biến x, index = 9 giá trị là **2.44357811** là đã vượt qua biên trên đã cho trước là 2 ở trên đè thì sẽ bằng -2. Hiểu rõ hơn sự thay đổi này xem những ô được in đậm trong Bảng 5 phía dưới:

Bảng 5. Các cá thể sau khi được kiểm tra ranh giới

Index	x	y	z
0	0.01831771	-0.95450176	-0.07420584
1	-2	-2	1.29357125
2	-0.21671154	0.42137008	0.12696972
3	-0.25529128	0.06973586	0.04292713
4	0.07093222	-0.26234487	0.21619166
5	0.10262446	0.00257076	0.12621516
6	-0.15790724	-1.02718838	0.82213052
7	-2	-1.42660959	2
8	0.69234925	2	0.64428492
9	-2	2	2

Kiểm tra biên xong sẽ sắp xếp lại các vị trí cá thể và kiểm tra nếu vị trí con đầu đàn thỏa điều kiện ra kết quả tốt nhất thì sẽ được thay thế, vì vậy kết quả sẽ thu được là Cá thể tốt nhất: [0.10262446, 0.00257076, 0.12621516] và fitness = 0.00616894. Hoàn tất vòng lặp đầu tiên thu được cá thể tốt nhất là cá thể được in đậm trong bảng còn cá thể bị thay thế thì sẽ được in nghiêng kết quả như Bảng 6 sau:

Bảng 6. Tập hợp các cá thể sau khi hoàn thành vòng lặp 1. Dòng in đậm là cá thể mới được thêm vào

Index	x	y	z	fitness
0	0.10262446	0.00257076	0.12621516	0.006168942
1	-0.25529128	0.06973586	0.04292713	0.024463214
2	0.07093222	-0.26234487	0.21619166	0.043879348
3	-0.21671154	0.42137008	0.12696972	0.107117888
4	0.01831771	-0.95450176	-0.07420584	0.456566402
5	-0.15790724	-1.02718838	0.82213052	0.648519315
6	0.69234925	2	0.64428492	2.228966338
7	-2	-1.42660959	2	3.017607461
8	-2	-2	1.29357125	3.612221096
9	-2	2	2	4

Thực hiện lặp lại (trong ví dụ này 100 vòng lặp). Các cá thể sau 100 vòng lặp như ở Bảng 7

Bảng 7. Tập hợp các cá thể sau 100 vòng lặp

Index	x	y	z	fitness
0	-1.515022E-42	7.665830E-42	-1.219732E-41	5.494333E-83
1	9.246556E-42	2.692784E-42	2.269989E-41	1.180060E-82
2	1.968101E-41	9.903141E-42	-1.086236E-41	1.978153E-82
3	2.862307E-41	-2.576713E-42	3.985507E-41	5.411508E-82
4	-2.872071E-41	4.346358E-42	6.835223E-41	1.063076E-81
5	-8.803673E-41	1.629523E-42	3.229520E-41	2.758646E-81
6	-3.591634E-38	2.369224E-38	-2.498157E-40	7.106660E-76

Index	x	y	z	fitness
7	2.620179E-29	9.175567E-40	7.037336E-37	2.288446E-58
8	-3.461296E-19	-3.743543E-19	-2.236614E-18	9.437459E-37
9	-2.624466E-12	2.161399E-21	-7.966598E-21	2.295940E-24

Dòng đầu tiên của Bảng 7 sẽ cho lời giải tối ưu của bài toán.

4. ỨNG DỤNG TRONG BÀI TOÁN ẨN TẬP PHỔ BIẾN

Bài toán (FIPP - Frequent Itemset Privacy Preserving) bảo vệ quyền riêng tư của các bộ dữ liệu giao tác bằng cách ẩn các tập phổ biến nhạy cảm trong dữ liệu và giữ nguyên tính toàn vẹn của các tập phổ biến không nhạy cảm. Phương pháp này thực hiện biến đổi cơ sở dữ liệu để loại bỏ hoặc thay thế các mục trong các tập phổ biến nhạy cảm bằng cách ẩn các tập phổ biến nhạy cảm trong dữ liệu và giữ nguyên tính toàn vẹn của các tập phổ biến không nhạy cảm. Tóm lại, FIPP là một phương pháp ẩn dữ liệu hiệu quả giúp bảo vệ quyền riêng tư của các bộ dữ liệu giao tác.

Định nghĩa 1. (Bài toán ẩn tập phổ biến). Cho trước một cơ sở dữ liệu giao tác F có tập hạng mục $I = \{i_1, i_2, \dots, i_r\}$, một độ hỗ trợ tối thiểu ω và một tập các tập phổ biến nhạy cảm SIs. Thực hiện biến đổi F thành F' sao cho $\text{sup}_{F'}(X) < \omega \forall X \in B$.

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 θ .

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 .

Định nghĩa 5. Sử dụng công thức Chim Hồng Hạc đơn giản, ta có thể tính toán được số lượng giao tác cần phải ẩn bằng cách nhân tỷ lệ minsup mong muốn với số lượng giao tác trong tập dữ liệu. Giả sử chúng ta muốn ẩn các tập phổ biến nhạy cảm có độ phổ biến thấp hơn 20% ($\text{minsup} = 20\%$) trong tập dữ liệu này. Số lượng giao tác cần ẩn = $\text{minsup} \times \text{số lượng giao tác trong tập dữ liệu} = 20\% \times 100 = 20$ giao tác

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. Cách tiếp cận tốt nhất để ẩn tập phổ biến phụ thuộc vào bối cảnh và mục đích sử dụng. Như vậy, bài toán ẩn tập phổ biến có thể quy về bài toán tối ưu với hàm mục tiêu

$$f = G_1\alpha + G_2\theta + G_3\gamma$$

trong đó, G1, G2, G3 là các trọng số tương ứng với các thông số không mong muốn. Thông thường G1 sẽ được đặt lớn hơn nhiều so với G2 và G3 với mong muốn có thể ẩn hoàn toàn các tập nhạy cảm.

Áp dụng phương pháp tối ưu chim hồng hạc cho bài toán ẩn tập nhạy cảm, công trình này đề xuất một phương pháp mới để giải bài toán theo hướng metaheuristic. Mỗi cá thể chim hồng hạc sẽ chứa biến Dim là không gian tìm kiếm. Mỗi cá thể chim hồng hạc sẽ chứa thông tin về Tid (Transaction ID) cần hiệu chỉnh hoặc xóa bỏ.

Dim có thể tính theo công thức:

$$Dim = \varepsilon_1 - \omega \frac{Max_Sup(B_i) - |D|}{\omega}$$

Trong đó:

ε_1 : random ngẫu nhiên -1 hoặc 1.

Max_sup (Bi): độ hỗ trợ lớn nhất trong số tất cả các tập nhạy cảm.

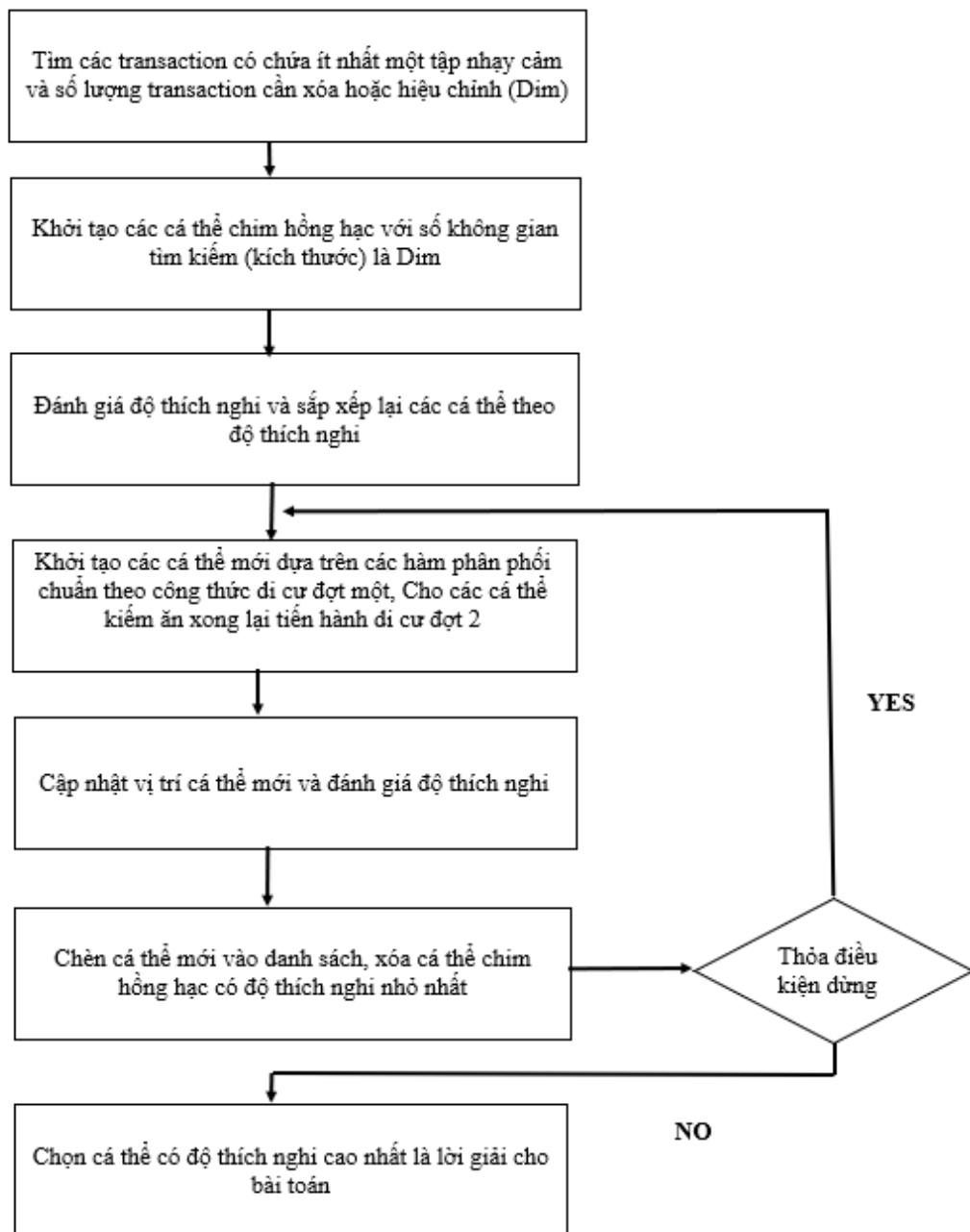
ω : độ hỗ trợ tối thiểu.

|D|: số lượng các giao tác có trong tập dữ liệu gốc.

Sau khi tính được số lượng không gian tìm kiếm trong một cá thể chim hồng hạc, các cá thể sẽ được khởi tạo một cách ngẫu nhiên dựa trên các thông tin các transaction có chứa ít nhất 1 tập nhạy cảm. Các cá thể được khởi tạo sẽ được đánh giá độ thích nghi sau đó và sắp xếp lại theo độ thích nghi.

Trong mỗi vòng lặp, một cá thể chim hồng hạc mới được phát sinh dựa trên hình thức di cư đột một tiếp đến kiểm ăn bằng cách di chuyển hai chân và quét mỏ sau đó sẽ tiến hành di cư đột 2 kiểm tra ranh giới và tính độ thích nghi, thông tin của 3 chiều không gian tìm kiếm được lựa chọn là con tốt nhất tương ứng với cá thể đầu đàn. Cá thể tốt nhất đã được sinh ra và cho chúng bắt buộc di chuyển ngẫu nhiên bằng cách nhân với một số ngẫu nhiên có thông số chuẩn.

Cá thể này sẽ được đánh giá độ thích nghi và chèn vào trong danh sách các cá thể đúng vị trí ứng với độ thích nghi của nó. Cá thể có độ thích nghi thấp nhất sẽ bị loại bỏ. Phương pháp ẩn tập phổ biến sử dụng tối tìm kiếm chim hồng hạc được biểu thị bởi sơ đồ ở Hình 4.



Hình 4. Sơ đồ phương pháp giải bài toán ẩn tập phổ biến sử dụng FSA

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này trình bày thuật toán tối ưu hóa tìm kiếm chim hồng hạc (Flamingo Search Algorithm-FSA) một phương pháp tối ưu hóa metaheuristic lấy cảm hứng từ đàn chim hồng hạc di cư kiếm ăn hoạt động trong tự nhiên. Với khả năng giải quyết các bài toán tối ưu đa mục tiêu. Trong bài báo cũng trình bày phương pháp để xuất ứng dụng thuật toán tối ưu tìm kiếm chim hồng hạc 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 không kém các phương pháp sử dụng tối ưu bầy đàn hay thuật giải di truyền trước đó.

Hướng phát triển tiếp theo cho thuật toán chim hồng hạc có thể bao gồm việc phát triển các biến thể của thuật toán, như thuật toán chim lửa hay thuật toán chim đại bàng, để tăng tính linh hoạt và khả năng giải quyết các bài toán khác nhau. Ngoài ra, cần nghiên cứu các phương pháp mới để cải thiện khả năng đánh giá hiệu quả của các cá thể mới, tăng tốc quá trình tiến hóa và giảm độ phức tạp tính toán nhằm cải thiện hiệu suất và độ chính xác của thuật toán. Sự tiến bộ trong thuật toán này sẽ đóng góp tích cực cho sự phát triển của nhiều lĩnh vực ứng dụng khác nhau.

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

TÀI LIỆU THAM KHẢO

1. Wang Zhiheng And Liu Jianhua,2021, Flamingo Search Algorithm: A New Swarm Intelligence Optimization Algorithm, Digital Object Identifier 10.1109/ACCESS.2021.3090512, 88564-88582.
2. J Q. Zhang, Y. Guo, Y. Wang, and X. Liu, “A discrete whale optimization algorithm and application,” *J. Univ. Electron. Sci. Technol. China*, vol. 49, no. 4, pp. 622–630, Jul. 2020, doi: 10.12178/1001 0548.2019116
3. S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,” *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art.no.103541, doi: 10.1016/j.engappai.2020.103541.
4. J. Xue and B. Shen, “A novel swarm intelligence optimization approach: Sparrow search algorithm,” *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020, doi: 10.1080/21642583.2019.1708830.
5. BØchet, M. Rendón-martos, M. Rendón, J. A. Amat, A. R. Johnson, and M. Gauthier-Clerc, “Global economy interacts with climate change to jeopardize species conservation: The case of the greater flamingo in the Mediterranean and West Africa,” *Environ. Conservation*, vol. 39, no. 1, pp. 1–3, Mar. 2012, doi: 10.1017/S0376892911000488.
6. M. Gao, H. Tang, and P. Zhang, “Survey of path planning technologies for robot swarm,” *J. Nat. Univ. Defense Technol.*, vol. 43, no. 1, pp. 127–138, Feb. 2021, doi: 10.11887/j.cn.202101017

ABSTRACT

FLAMINGO SEARCH ALGORITHM OPTIMIZES AND APPLICATION IN HIDING FREQUENT ITEMSET PROBLEMS

Le Huynh Duc¹, Nguyen Thi Hang¹, Dinh Nguyen Trong Nghia^{1,*}

¹*Ho Chi Minh City University of Food Industry*

*Email: nghiadnt@hufi.edu.vn

This paper introduces an intelligent swarm optimization algorithm, named "Flamingo Search Algorithm" (FSA), based on the migratory and foraging behavior of flamingos. This algorithm is capable of finding optimal solutions to optimization problems quickly and efficiently. The FSA uses parameters such as the maximum number of iterations, the size of the flamingo population, the search range limit, and the number of migratory and foraging flocks of flamingos. The article also proposes to apply FSA to the problem of hidden frequent sets, to help find the optimal solution and minimize the risks related to competition among individuals. The FSA algorithm distributes the objects evenly on the search space and moves to locations with a lot of useful information, helping to find the best solution.

Keywords: Flamingo Search Algorithm, FSA, PPDM, hiding frequent itemset