**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT**

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo of a university of transport technology

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**TRÍ TUỆ NHÂN TẠO**

**Tên đề tài:**

**THUẬT TOÁN TỐI ƯU HÓA BẦY ĐÀN**

**THUÂT TOÁN TỐI ƯU HÓA BẦY ONG**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN:** | **ĐỖ BẢO SƠN** |
| **NHÓM SINH VIÊN THỰC HIỆN:** | **1. ĐẶNG NGỌC HIỆU**  **2. KIỀU MINH QUÂN**  **3. ĐỖ QUỐC TRƯỜNG**  **4. NGUYỄN HẢI DƯƠNG**  **5. BẠC ANH TUẤN** |
| **LỚP:** | **73DCTT23** |

**HÀ NỘI 04-2022**

**Bảng phân công nhiệm vụ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ và Tên | Mã sinh viên | Nhiệm vụ | Ghi chú |
| 1 | Đặng Ngọc Hiệu |  | Tìm hiểu nội dung, DEMO |  |
| 2 | Kiều Minh Quân | 73DCTT22202 | Tìm hiểu nội dung, Word |  |
| 3 | Đỗ Quốc Trường | 73DCTT22120 | Tìm hiểu nội dung, Word |  |
| 4 | Nguyễn Hải Dương |  | Tìm hiểu nội dung, Powerpoint |  |
| 5 | Bạc Anh Tuấn |  | Tìm hiểu nội dung, DEMO |  |

**Mục lục**

[LỜI MỞ ĐẦU 4](#_Toc184826311)

[CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO 5](#_Toc184826312)

[1. Khái niệm 5](#_Toc184826313)

[2. Lịch sử phát triển 5](#_Toc184826314)

[3. Ưu điểm và nhược điểm của trí tuệ nhân tạo 6](#_Toc184826315)

[4. Các thuật toán phổ biến 7](#_Toc184826316)

[CHƯƠNG 2: THUẬT TOÁN TỐI ƯU BẦY ĐÀN (PSO – PARTICLE SWARM OPTIMIZATION) 9](#_Toc184826317)

[1. Giới thiệu 9](#_Toc184826318)

[2. Lý do ra đời 10](#_Toc184826319)

[3. Lợi ích của thuật toán bầy đàn 10](#_Toc184826320)

[4. Nguyên lý hoạt động và các bước của thuật toán 12](#_Toc184826321)

[5. Ứng dụng của thuật toán PSO 19](#_Toc184826322)

[CHƯƠNG 3: THUẬT TOÁN TỐI ƯU BẦY ONG (ARTIFICIAL BEE COLONY) 21](#_Toc184826323)

[1. Giới thiệu 21](#_Toc184826324)

[2. Lý do ra đời 21](#_Toc184826325)

[3. Lợi ích của thuật toán bầy ong 22](#_Toc184826326)

[4. Nguyên lý hoạt động và các bước thuật toán 22](#_Toc184826327)

[5. Ứng dụng của thuật toán ABC 28](#_Toc184826328)

[CHƯƠNG 4: KẾT LUẬN 30](#_Toc184826329)

[Tài liệu tham khảo 31](#_Toc184826330)

# **LỜI MỞ ĐẦU**

Trong thực tế và khi xây dựng các hệ thông tin, ta thường gặp các bài toán tối ưu tổ hợp (TƯTH). Trong đó phải tìm các giá trị cho các biến rời rạc để làm cực trị hàm mục tiêu nào đó. Đa số các bài toán này thuộc lớp NP-khó. Trừ các bài toán cỡ nhỏ có thể tìm lời giải bằng cách tìm kiếm vét cạn, còn lại thì thường không thể tìm được lời giải tối ưu. Đối với các bài toán cỡ lớn không có phương pháp giải đúng, đến nay người ta vẫn dùng các cách tiếp cận sau:

1) Tìm kiếm heuristic để tìm lời giải đủ tốt;

2) Tìm kiếm cục bộ để tìm lời giải tối ưu địa phương;

3) Tìm lời giải gần đúng nhờ các thuật toán mô phỏng tự nhiên như: mô phỏng luyện kim, giải thuật di truyền, tối ưu bầy đàn,…

Các thuật toán thông minh lấy cảm hứng từ tự nhiên đã chứng minh được tiềm năng to lớn trong việc giải quyết những bài toán này, nhờ khả năng mô phỏng hành vi của các loài động vật trong tự nhiên để tìm kiếm lời giải tối ưu. Trong số đó, Thuật toán tối ưu Bầy đàn (PSO - Particle Swarm Optimization) và Thuật toán Bầy ong nhân tạo (Artificial Bee Colony - ABC) là hai đại diện tiêu biểu.

Thuật toán tối ưu Bầy đàn là một lĩnh vực nghiên cứu trong khoa học máy tính và kỹ thuật lấy cảm hứng từ hành vi tập thể của các sinh vật như kiến, chim, cá và côn trùng. Những thuật toán này được dựa trên cách mà các sinh vật trong bầy hoạt động cùng nhau để giải quyết các vấn đề phức tạp.

Thuật toán Bầy ong nhân tạo là một phương pháp tối ưu hóa lấy cảm hứng từ hành vi kiếm ăn của bầy ong. ABC được phát triển bởi Derviş Karaboğa vào năm 2005 và được áp dụng trong nhiều lĩnh vực khác nhau.

Việc nghiên cứu hai thuật toán này không chỉ giúp hiểu rõ hơn về cách chúng hoạt động, mà còn mở ra các hướng ứng dụng thực tiễn trong nhiều lĩnh vực như logistics, trí tuệ nhân tạo, và khoa học dữ liệu. Trong bài viết này, chúng ta sẽ tìm hiểu chi tiết về cơ chế hoạt động, ưu điểm, nhược điểm và các ứng dụng nổi bật của cả hai thuật toán.

# **CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO**

**1. Khái niệm**

Trí tuệ nhân tạo (Artificial Intelligence - AI) là một lĩnh vực nghiên cứu trong khoa học máy tính và kỹ thuật nhằm tạo ra các hệ thống và máy móc có khả năng thực hiện các nhiệm vụ thường đòi hỏi trí tuệ của con người, giúp máy móc suy nghĩ và đưa ra quyết định như con người (“The science and engineering of making intelligent machines.” - Khoa học và kỹ thuật chế tạo máy móc thông minh - John McCarthy – 1956)

**2. Lịch sử phát triển**

- 1943: McCulloch & Pitts trình bày công trình nghiên cứu đầu tiên về AI, đề xuất mô hình các nơ-ron nhân tạo 2 trạng thái

- 1950: Khái niệm về TTNT lần đầu tiên được Turing đề cập trong bài báo "Computing Machinery and Intelligence”

- 1956: Workshop đầu tiên (diễn ra trong 2 tháng) ở Dartm bàn về lĩnh vực TTNT, khái niệm TTNT được thừa nhận

- 1952-1969: Các thành tựu ban đầu trong TTNT

- 1950s: Các chương trình TTNT đầu tiên

* + Chương trình chơi cờ của Samuel
  + Chương trình lý luận logic của Newell & Simon
  + Chương trình chứng minh các định lý hình học của Gelernter

- 1965: Robinson đề cử giải thuật hoàn chỉnh cho việc suy diễn logic

- 1966-1973:

* Các nhà nghiên cứu về TTNT nhận ra khó khăn về độ phức tạp tính to
* Gần như là không còn các nghiên cứu về các mạng nơ-ron nhân tạo

- 1969-1979: Sự hình thành và phát triển ban đầu của các hệ thống dựa trên tri thức

- 1980: TTNT trở thành một ngành công nghiệp (các hệ thống, chương trình TTNT dùng trong thương mại)

- 1980-1988: Sự xuất hiện bùng nổ của các hệ chuyên gia

- 1986: Các mạng nơ-ron nhân tạo xuất hiện trở lại, trở nên phổ biến

- 1987: TTNT trở thành một lĩnh vực khoa học

- 1995: Sự xuất hiện của các tác tử thông minh

- Thế kỷ 21: Big Data, Machine learning, Deep Learning và ứng dụng thương mại rộng rãi của Trí tuệ nhân tạo.

**3. Ưu điểm và nhược điểm của trí tuệ nhân tạo**

a. Ưu điểm

* Tăng hiệu quả và năng suất làm việc:

AI có thể thực hiện các nhiệm vụ lặp đi lặp lại với tốc độ nhanh và độ chính xác cao, giúp giảm thời gian và chi phí. Ví dụ: dây chuyền sản xuất tự động hoặc xử lý dữ liệu quy mô lớn.

* Khả năng xử lý dữ liệu khổng lồ:

AI có thể phân tích và xử lý lượng dữ liệu lớn mà con người không thể quản lý hiệu quả, từ đó đưa ra các quyết định hoặc dự đoán có giá trị.

* Cải thiện chất lượng cuộc sống:

AI góp phần nâng cao chất lượng cuộc sống thông qua các ứng dụng như trợ lý ảo, nhà thông minh, thiết bị hỗ trợ y tế và công nghệ hỗ trợ người khuyết tật.

* Tăng độ chính xác trong các ngành quan trọng:

Trong y tế, AI hỗ trợ chẩn đoán bệnh sớm và chính xác hơn. Trong tài chính, nó phát hiện gian lận một cách hiệu quả.

* Khả năng hoạt động liên tục:

AI có thể làm việc 24/7 mà không cần nghỉ ngơi, giúp duy trì hoạt động không gián đoạn trong nhiều lĩnh vực như dịch vụ khách hàng hoặc giám sát an ninh.

* Ứng dụng đa dạng:

AI có thể được sử dụng trong nhiều lĩnh vực khác nhau như giáo dục, giao thông, năng lượng, công nghiệp, và nghiên cứu khoa học.

b. Nhược điểm

* Chi phí phát triển cao:

Việc phát triển, triển khai và duy trì các hệ thống AI đòi hỏi nguồn lực tài chính lớn và đội ngũ chuyên gia có trình độ cao.

* Thiếu linh hoạt trong tư duy sáng tạo:

AI chỉ có thể thực hiện các nhiệm vụ trong phạm vi mà nó được lập trình hoặc học hỏi, và không thể thay thế hoàn toàn sự sáng tạo hay khả năng cảm nhận của con người.

* Nguy cơ mất việc làm:

Sự tự động hóa do AI mang lại có thể dẫn đến việc thay thế lao động con người, đặc biệt trong các công việc đơn giản hoặc lặp đi lặp lại.

* Rủi ro về đạo đức và quyền riêng tư:

AI có thể được sử dụng để theo dõi, thu thập thông tin cá nhân hoặc đưa ra quyết định thiếu công bằng, gây lo ngại về bảo mật và quyền con người.

* Khả năng mắc lỗi trong các tình huống phức tạp:

Mặc dù rất chính xác trong nhiều trường hợp, AI vẫn có thể thất bại khi gặp phải các tình huống bất thường mà nó chưa được đào tạo để xử lý.

* Phụ thuộc vào dữ liệu:

Hiệu suất của AI phụ thuộc vào chất lượng và số lượng dữ liệu. Nếu dữ liệu sai hoặc thiếu sót, kết quả mà AI đưa ra cũng sẽ không chính xác.

* Nguy cơ lạm dụng công nghệ:

AI có thể bị sử dụng vào mục đích xấu như tấn công mạng, sản xuất thông tin giả mạo (deepfake), hoặc phát triển vũ khí tự động.

**4. Các thuật toán phổ biến**

- Thuật toán Di truyền (Genetic Algorithm - GA): là một phương pháp tối ưu hóa và tìm kiếm dựa trên nguyên lý chọn lọc tự nhiên và tiến hóa sinh học của Charles Darwin. Được giới thiệu bởi John Holland vào những năm 1970, GA mô phỏng quá trình tiến hóa trong tự nhiên để tìm kiếm lời giải tối ưu cho các bài toán phức tạp.

- Thuật toán Dơi (Bat Algorithm - BA): là một thuật toán tối ưu hóa lấy cảm hứng từ hành vi định vị bằng sóng siêu âm (echolocation) của loài dơi. Được đề xuất bởi Xin-She Yang vào năm 2010, thuật toán này mô phỏng cách dơi sử dụng sóng siêu âm để phát hiện các chướng ngại vật và tìm kiếm con mồi, từ đó tìm ra giải pháp tối ưu trong không gian tìm kiếm.

- Thuật toán tối ưu bầy đàn (PSO - Particle Swarm Optimization): là một lĩnh vực nghiên cứu trong khoa học máy tính và kỹ thuật lấy cảm hứng từ hành vi tập thể của các sinh vật như kiến, chim, cá và côn trùng. Dựa trên hành vi kiếm ăn của chim và cá, trong đó mỗi phần tử (particle) di chuyển trong không gian tìm kiếm và điều chỉnh vị trí của mình dựa trên vị trí tốt nhất của mình và của cả bầy.

- Thuật toán bầy ong nhân tạo (ABC - Artificial Bee Colony): là một phương pháp tối ưu hóa lấy cảm hứng từ hành vi kiếm ăn của bầy ong. ABC được phát triển bởi Derviş Karaboğa vào năm 2005 và được áp dụng trong nhiều lĩnh vực khác nhau. Dựa trên hành vi kiếm ăn của bầy ong, trong đó các loại ong thợ, ong quan sát và ong do thám cùng hợp tác để tìm kiếm và tối ưu hóa nguồn thức ăn.

# **CHƯƠNG 2: THUẬT TOÁN TỐI ƯU BẦY ĐÀN (PSO – PARTICLE SWARM OPTIMIZATION)**

**1. Giới thiệu**

Phương pháp tối ưu bầy đàn là một dạng của các thuật toán tiến hóa quần thể đã được biết đến trước đây như giải thuật di truyền (Genetic algorithm (GA)), thuật toán đàn kiến (Ant colony algorithm). Tuy vậy PSO khác với GA ở chỗ nó thiên về sử dụng sự tương tác giữa các cá thể trong một quần thể để khám phá không gian tìm kiếm. PSO là kết quả của sự mô hình hóa việc đàn chim bay đi tìm kiếm thức ăn cho nên nó thường được xếp vào các loại thuật toán có sử dụng trí tuệ bầy đàn. Được giới thiệu vào năm 1995 tại một hội nghị của IEEE bởi James Kennedy và kỹ sư Russell C.Eberhart. Thuật toán có nhiều ứng dụng quan trọng trong tất cả các lĩnh vực mà ở đó đòi hỏi phải giải quyết các bài toán tối ưu hóa. Để hiểu rõ thuật toán PSO hãy xem một ví dụ đơn giản về quá trình tìm kiếm thức ăn của một đàn chim. Không gian tìm kiếm thức ăn lúc này là toàn bộ không gian ba chiều mà chúng ta đang sinh sống. Tại thời điểm bắt đầu tìm kiếm cả đàn bay theo một hướng nào đó, có thể là rất ngẫu nhiên. Tuy nhiên sau một thời gian tìm kiếm một số cá thể trong đàn bắt đầu tìm ra được nơi có chứa thức ăn. Tùy theo số lượng thức ăn vừa tìm kiếm, mà cá thể gửi tín hiệu đến các các cá thể khác đang tìm kiếm ở vùng lân cận. Tín hiệu này lan truyền trên toàn quần thể. Dựa vào thông tin nhận được mỗi cá thể sẽ điều chỉnh hướng bay và vận tốc theo hướng về nơi có nhiều thức ăn nhất. Cơ chế truyền tin như vậy thường được xem như là một kiểu hình của trí tuệ bầy đàn. Cơ chế này giúp cả đàn chim tìm ra nơi có nhiều thức ăn nhất trên không gian tìm kiếm vô cùng rộng lớn. Như vậy đàn chim đã dùng trí tuệ, kiến thức và kinh nghiệm của cả đàn để nhanh chóng tìm ra nơi chứa thức ăn. Bây giờ chúng ta tìm hiểu làm cách nào mà một mô hình trong sinh học như vậy có thể áp dụng trong tính toán và sinh ra thuật toán PSO mà ta từng nhắc đến. Việc mô hình hóa này thường được gọi là quá trình phỏng sinh học (bioinspired) mà chúng ta thường thấy trong các ngành khoa học khác. Một thuật toán được xây dựng dựa trên việc mô hình hóa các quá trình trong sinh học được gọi là thuật toán phỏng sinh học (bioinspired algorithms).

**2. Lý do ra đời**

Thuật toán bầy đàn ra đời với mục đích mô phỏng và tận dụng các hành vi tập thể của sinh vật trong tự nhiên để giải quyết các vấn đề phức tạp trong khoa học máy tính và kỹ thuật. Các sinh vật như kiến, ong, chim và cá đã tiến hóa qua hàng triệu năm để phát triển các hành vi tập thể hiệu quả và tối ưu. Các nhà nghiên cứu đã nhận ra rằng việc mô phỏng các hành vi này có thể giúp giải quyết các vấn đề phức tạp trong thực tế. Ngoài ra các hệ thống bầy đàn hoạt động theo cơ chế phân tán và tự tổ chức, không cần sự điều khiển tập trung. Điều này giúp hệ thống bầy đàn có khả năng linh hoạt, thích nghi và bền vững cao hơn so với các hệ thống điều khiển tập trung. Các thuật toán bầy đàn có khả năng tìm kiếm toàn cục tốt hơn và tránh bị kẹt trong các điểm tối ưu cục bộ nhờ vào sự tương tác giữa các phần tử trong bầy. Các thuật toán bầy đàn có cấu trúc đơn giản và dễ dàng mở rộng để giải quyết các bài toán lớn hơn. Chúng có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ tối ưu hóa công nghiệp, điều khiển robot, lập lịch, đến học máy và xử lý dữ liệu lớn. Việc áp dụng các thuật toán bầy đàn vào các vấn đề thực tế có thể giúp cải thiện hiệu suất, tăng hiệu quả và giảm chi phí trong các quy trình và hệ thống.

**3. Lợi ích của thuật toán bầy đàn**

* Tối ưu hóa tốt:

Khả năng giải quyết những bài toán cực kì rối: Thuật toán bầy đàn, như Particle Swarm Optimization (PSO) và Ant Colony Optimization (ACO), đã cho thấy khả năng xuất sắc trong việc tìm kiếm giải pháp gần như tối ưu cho các bài toán tối ưu hóa đa mục tiêu. Những bài toán này thường yêu cầu rất kĩ về những kĩ năng yếu tố khác nhau, từ đó đưa ra giải pháp tốt nhất.

* Tính linh hoạt:

Đa dạng ứng dụng: Các thuật toán bầy đàn có thể được áp dụng cho nhiều loại bài toán khác nhau mà không cần phải lo lắng cho sự chính xác và độ phức tạp mà nó mang lại. Chúng có thể được điều chỉnh để tối ưu hóa các vấn đề trong khoa học máy tính, kỹ thuật, tài chính và còn nhiều lĩnh vực khác

* Cách thức hoạt động:

Trong thuật toán bầy đàn, mỗi cá thể đại diện cho một giải pháp tiềm năng và được biểu diễn trong lĩnh vực rộng lớn về không gian đa chiều. Mỗi lần khi vòng lặp, các cá thể chia sẻ thông tin về những gì mà đã tìm thấy những vị trí tốt nhất.

* Chi tiết ứng dụng:

PSO có thể được sử dụng để tối ưu hóa khoa học máy tính, thiết kế sản phẩm, hoặc trong việc tối ưu hóa danh mục đầu tư tài chính, cải thiện hơn trong quy trình sản xuất tạo ra sản phẩm. Điều này cho thấy tính linh hoạt cao của các thuật toán này.

Thuật toán bầy đàn rất dễ triển khai:

Cấu trúc đơn giản và trực quan: Các thuật toán bầy đàn như PSO và ACO thường có cấu trúc dễ hiểu làm cho người sử dụng cảm thấy yên tâm về sự chính xác cũng như giá trị mang lại, cho phép người dùng dễ dàng triển khai khi không cần có kiến thức chuyên sâu về toán học hoặc lập trình.

* Tính đáng tin cậy:

Các cá thể sống độc lập, thuật toán bầy đàn có khả năng duy trì hiệu suất ngay cả khi một số cá thể không hoạt động. Điều này có thể rất quan trọng trong các ứng dụng cần độ tin cậy cao, như trong lĩnh vực tự động hóa và điều khiển. Có một vài cá thể không thể thực hiện được nhiệm vụ, những cá thể khác có thể làm để đảm bảo rằng hệ thống không bị ảnh hưởng quá nhiều.

* Góp phần chia sẻ thông tin:

Thuật toán bầy đàn cho phép các cá thể trong bầy chia sẻ thông tin về các giải pháp mà chúng đã tìm thấy. Điều này tạo ra một quá trình học hỏi tập thể nơi mà mỗi cá thể có thể cải thiện dựa trên kinh nghiệm của cá thể khác. Việc này giúp tối ưu hóa hiệu suất tìm kiếm vì các cá thể sẽ không chỉ phụ thuộc vào kinh nghiệm của chính mình mà còn có thể học hỏi từ những lần đột phá và thành công của người khác để có thể trau dồi trở nên tốt nhất.

* Giải quyết nhiều mục tiêu:

Nhiều thuật toán bầy đàn được thiết kế để giải quyết các bài toán tối ưu hóa đa mục tiêu. Điều này có nghĩa là bạn có thể đồng thời tối ưu hóa nhiều yếu tố như chi phí, hiệu suất và độ tin cậy. Trong một vài trường hợp như là quản lí dự án bạn có thể tối ưu hóa được chi phí thời gian và chất lượng của công việc. Các thuật toán bầy đàn này có thể giúp cho bạn cân bằng được những yếu tố trên.

* Khả năng thích ứng:

Thuật toán bầy đàn có khả năng thích ứng với sự thay đổi trong môi trường hoặc dữ liệu. Việc này đóng vai trò vô cùng quan trọng trong các tình huống mà điều kiện và yêu cầu của nó có thể thay đổi một cách nhanh chóng như trong các hệ thống quản lí lưu lượng giao thông. Vì một vài lý do mà dữ liệu nó thay đổi liên tục thuật toán này có thể điều chỉnh những hành vi của cá thể trong bầy để tìm được giải pháp tốt nhất.

* Tính tương đối cao:

Thuật toán bầy đàn có thể tìm kiếm trong không gian giải pháp mà không cần quá nhiều thông tin ban đầu. Điều này đã làm cho chúng hoạt động tốt trong những trường hợp bài toán có quá ít dữ liệu và không có thông tin lành mạch rõ ràng. Vì vậy bạn không cần phải xác định hướng đi ban đầu, mà có thể để khám phá và cải thiện dần dần thông qua hành vi của thuật toán này.

* Chi phí thấp:

So với nhiều phương thức khác nhau thì thuật toán bầy đàn có thể yêu cầu ít dữ liệu tính toán hơn trong nhiều trường hợp. Điều này đã góp phần làm cho giúp tiết kiệm được một khoản cho những chi phí về các dự án lớn, vì không cần phải đầu tư quá nhiều vào những phần như phần cứng hay phần mềm phức tạp. Vì thế bạn có thể tận dụng những công cụ cần thiết và kỹ thuật đơn giản mà hiệu quả để giải quyết những bài toán phức tạp khó khăn.

**4. Nguyên lý hoạt động và các bước của thuật toán**

a. Nguyên lý hoạt động cơ bản

Nguyên lý hoạt động cơ bản của thuật toán bầy đàn dựa trên việc mô phỏng hành vi của các nhóm sinh vật trong tự nhiên, như đàn chim, đàn cá hoặc đàn kiến. Dưới đây là những điểm chính trong nguyên lý hoạt động của thuật toán này:

- Tính phi tập trung: Trong bầy sẽ không có bất kỳ cá thể nào là trung tâm điều khiển. Mỗi cá thể trong bầy sẽ tự điều chỉnh hành động của mình và tương tác với các cá thể khác dựa vào đặc điểm của bầy. Từ đó giúp cho bầy linh hoạt và dễ dàng thích nghi với các môi trường khác nhau.

- Khả năng thích nghi và phát triển: Các cá thể trong đàn có thể tự học hỏi và thay đổi hành động của mình dựa trên sự tác động của môi trường để cải thiện và tối ưu hiệu suất của bầy.

- Sự tự tổ chức: Các hành động nhỏ đơn giản của mỗi cá thể tạo ra một hệ thống phức tạp mà không cần chỉ đạo.

Ví dụ: Những con kiến đầu tiên tìm đường ngẫu nhiên sau đó để lại Pheromone trên đường đi để đánh dấu, sau đó những con kiến sau có thể tìm thức ăn và đường về tổ nhờ Pheromone để lại từ những con kiến trước đó.

-Phản hồi tích cực và tiêu cực: Các hành động có ích cho bầy sẽ được mở rộng và phát triển, còn các hành gây hại cho đàn sẽ bị hạn chế và giảm thiểu.

Ví dụ: Khi kiến tìm thấy thức ăn thì sẽ có nhiều con kiến đi trên một cung đường, lượng Pheromone sẽ tăng lên sẽ làm thu hút những con kiến khác.

-Tính đồng bộ: Những cá thể trong bầy sẽ hành động giống nhau và có thể phối hợp dễ dàng.

Ví dụ: Đàn cá sẽ bơi theo bày và có một hình dáng nhất định để tránh sự truy đuổi của những kẻ thù khác.

-Sự đa dạng: là những hành động và đặc điểm khác nhau của các cá thể trong bầy tạo nên nhiều cách giải quyết một vấn đề và tăng khả năng thích nghi và sống sót của bầy.

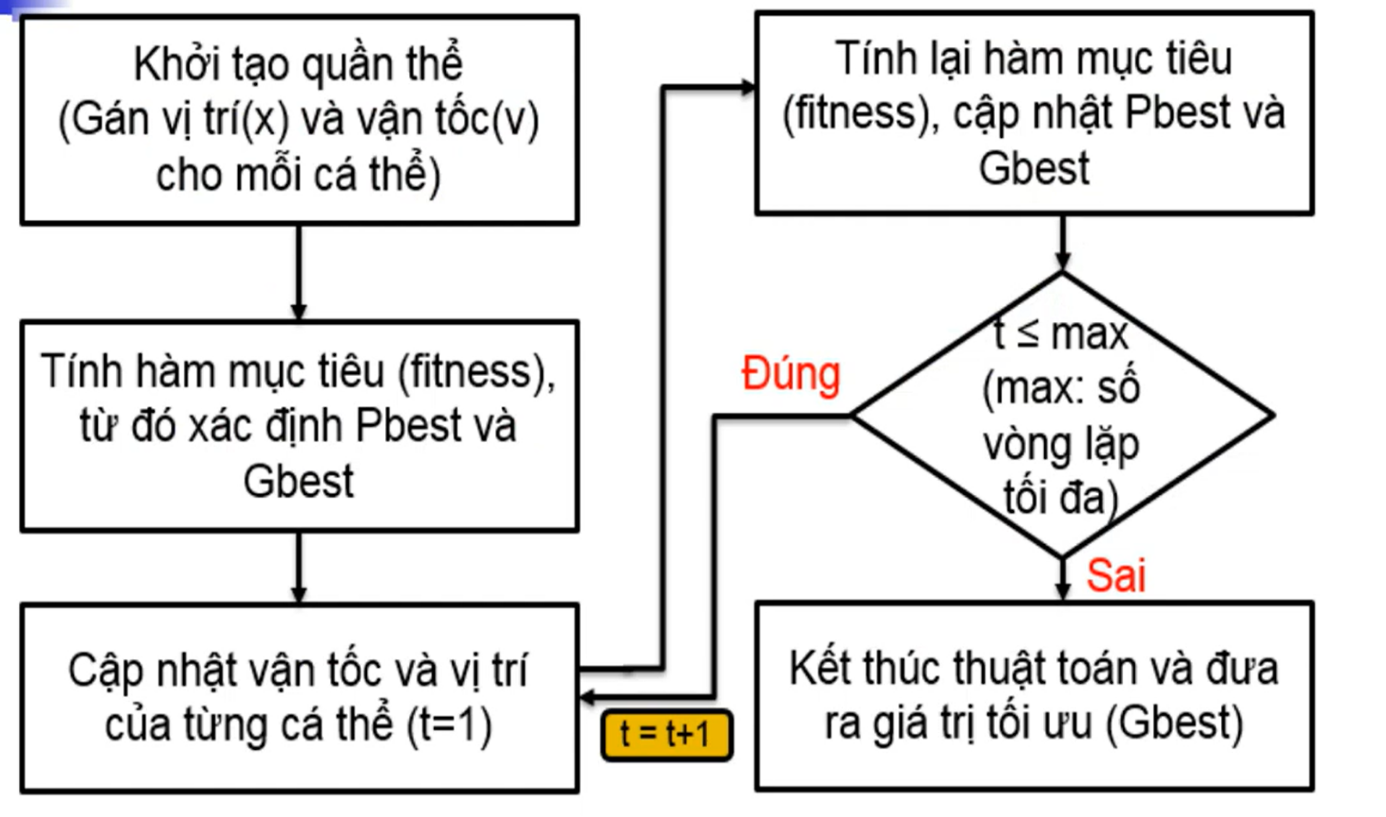
Ví dụ: Khi di chuyển, đàn chim sẽ có một vài con bay cao hơn và có công việc báo hiệu các mối nguy hiểm, còn những con chim ở dưới thấp sẽ đảm nhiệm công việc tìm kiếm thức ăn.

-Khả năng trao đổi thông tin: Các cá thể trong bầy sẽ có những hành động để trao đổi thông tin với nhau nhằm tăng khả năng tương tác giúp cho công việc đạt hiệu quả cao. Đây chính là phương pháp giúp tối ưu hóa việc tìm kiếm dữ liệu.

Ví dụ: Ong thợ sẽ thực hiện một loạt chuyển động gọi là “điệu nhảy lắc lư” để hướng dẫn những con ong thợ khác biết vị trí của phấn hoa.

- Khả năng tự thích ứng: Đó chính là khả năng phân chia lại và gánh vác nhiệm vụ khi một cá thể nào đó rủi ro và không thể hoạt động được nữa. Điều đó giúp cho bầy tiếp tục duy trì hoạt động và không bị gián đoạn, khả năng này giúp cho bầy trở nên bền vững và thích nghi tốt với sự khắc nghiệt và thử thách của môi trường. Ví dụ: Nếu trong bầy có một con ong thợ chết đi thì các con ông khác sẽ nhanh chóng tự phân chia lại nhiệm vụ với nhau. Những nguyên lý trên là cơ sở cho các thuật toán tối ưu bầy đàn và giúp cho thuật toán phát triển mạnh mẽ và mang lại hiệu quả cao cho việc giải quyết vấn đề phức tạp.

b. Các bước thuật toán

* + Theo dạng sơ đồ

Sơ đồ chung của thuật toán bầy đàn

Từ thuật toán trên ta có thể rút ra các bước giải quyết một bài toán ứng dụng với thuật toán bầy đàn:

*Bước 1: Khởi tạo*

* Khởi tạo quần thể: Ban đầu, tạo ra một quần thể gồm các hạt (particles) với các vị trí và vận tốc ngẫu nhiên trong không gian tìm kiếm. Các vị trí ban đầu này đại diện cho các giải pháp ngẫu nhiên của bài toán.

*Bước 2: Đánh giá và cập nhật thông tin cá thể*

* Đánh giá hàm mục tiêu: Tính toán giá trị hàm mục tiêu (fitness value) cho mỗi hạt dựa trên vị trí hiện tại của chúng trong không gian tìm kiếm. Hàm mục tiêu này phản ánh mức độ tốt của giải pháp.
* Lưu trữ vị trí tốt nhất của từng hạt (pbest): Mỗi hạt lưu trữ vị trí tốt nhất mà nó từng đạt được cho đến hiện tại (pbest). Nếu vị trí hiện tại của hạt có giá trị hàm mục tiêu tốt hơn vị trí tốt nhất trước đó, cập nhật pbest.
* Lưu trữ vị trí tốt nhất toàn cục (gbest): Toàn bộ quần thể lưu trữ vị trí tốt nhất mà bất kỳ hạt nào từng đạt được (gbest). Nếu có hạt nào có giá trị hàm mục tiêu tốt hơn gbest hiện tại, cập nhật gbest.

*Bước 3: Cập nhật vận tốc và vị trí*

* Cập nhật vận tốc: Mỗi cá thể sẽ điều chinh vận tốc của mình theo các cá thế có giá trị thích nghi tốt nhất như sau:

vi (t + 1) = w.vi(t) + c1.r1.(pbesti (t) – xi(t)) + c2.r2.(gbest (t) – xi(t))

Trong đó:

* + vi: Vận tốc hiện tại của hạt i.
  + w: Hệ số quán tính. (giúp cân bằng giữa khả năng khám phá không gian tìm kiếm và khai thác các giải pháp tốt)
  + c1, c2: Các hệ số học tập, ảnh hưởng đến mức độ mà hạt bị thu hút về vị trí pbest và gbest.
  + r1, r2: Các số ngẫu nhiên trong khoảng [0, 1], đảm bảo sự ngẫu nhiên trong quá trình cập nhật.
  + pbesti: Vị trí tốt nhất của hạt i.
  + xi: Vị trí hiện tại của hạt i.
  + gbest: Vị trí tốt nhất toàn cục.
* Cập nhật vị trí: Mỗi cá thể sẽ điều chinh vị trí của mình theo các cá thế có giá trị thích nghi tốt nhất như sau:

xi (t + 1) = xi(t) + vi(t+1)

*Bước 4: Kiểm tra điều kiện dừng*

* Kiểm tra điều kiện dừng: Xác định xem điều kiện dừng đã thỏa mãn chưa. Điều kiện dừng có thể là số vòng lặp tối đa hoặc khi giá trị hàm mục tiêu đạt được một ngưỡng nhất định. Nếu thỏa mãn, dừng quá trình; nếu không, quay lại bước 3.

Mỗi vòng lặp sau khi cá thể được cập nhật vị trí và đánh giá giá trị hàm mục tiêu, giá trị tốt nhất cá thể từng đạt được cũng được cập nhật. Giá trị vị trí tốt nhất mới của cá thể i trong vòng lặp thứ t+1 được định nghĩa :

pbesti (t+1)

Sau khi cập nhật các giá trị vị trí trong tập hợp P, việc xác định chỉ số g

cho vị trí có giá trị hàm mục tiêu tốt nhất sẽ hoàn thành một vòng lặp

của giải thuật PSO.

Bước 5: Kết quả

* Kết quả tối ưu: Sau khi điều kiện dừng được thỏa mãn, vị trí tốt nhất toàn cục (gbest) là giải pháp tối ưu được tìm thấy bởi thuật toán.

**Code Python:**

A computer screen shot of a computer code

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Chạy DEMO:**

* *Demo dạng 2D*

A white text with black numbers and a number

Description automatically generated with medium confidence

* *Demo dạng 3D*

A graph of a diagram

Description automatically generated with medium confidence

## **5. Ứng dụng của thuật toán PSO**

**a. Quản lý chuỗi cung ứng**

**Ví dụ**: Trong một công ty sản xuất, việc tối ưu hóa chuỗi cung ứng là rất quan trọng để giảm thiểu chi phí và đảm bảo sản phẩm được giao đúng thời hạn. Thuật toán PSO (Particle Swarm Optimization) có thể được sử dụng để tối ưu hóa các tuyến đường vận chuyển, lịch trình sản xuất và quản lý tồn kho. Chẳng hạn, PSO có thể giúp xác định lộ trình ngắn nhất để vận chuyển hàng hóa từ nhà máy đến các cửa hàng, đồng thời tối ưu hóa số lượng hàng tồn kho cần thiết để đáp ứng nhu cầu mà không gây lãng phí.

**b. Robot tự hành**

**Ví dụ**: Trong ngành công nghiệp tự động hóa, việc điều khiển nhóm robot di chuyển trong nhà máy là một nhiệm vụ phức tạp. Thuật toán PSO (Particle Swarm Optimization) có thể được áp dụng để điều khiển các robot tìm kiếm và chọn lộ trình tối ưu, tránh chướng ngại vật và hoàn thành nhiệm vụ một cách hiệu quả. PSO giúp các robot học từ các trải nghiệm trước đó để tìm kiếm con đường ngắn nhất và an toàn nhất.

c. Dự đoán thị trường tài chính

**Ví dụ**: Trong lĩnh vực tài chính, việc dự đoán giá cổ phiếu là một nhiệm vụ đầy thách thức. Thuật toán bầy đàn, chẳng hạn như PSO, có thể được sử dụng để tối ưu hóa các mô hình dự đoán giá cổ phiếu dựa trên dữ liệu lịch sử. PSO có thể điều chỉnh các tham số của mô hình để tăng cường độ chính xác của dự đoán, giúp các nhà đầu tư đưa ra quyết định hợp lý.

**d. Chẩn đoán bệnh**

**Ví dụ**: Trong y tế, thuật toán bầy đàn có thể được sử dụng để chẩn đoán bệnh dựa trên phân tích dữ liệu y tế. Chẳng hạn, PSO có thể được áp dụng để tối ưu hóa các tham số của mô hình học máy nhằm phân loại hình ảnh y tế và phát hiện các dấu hiệu bệnh tật. Điều này giúp bác sĩ chẩn đoán bệnh nhanh chóng và chính xác hơn.

**e. Tối ưu hóa mạng lưới giao thông**

**Ví dụ**: Trong quản lý giao thông, thuật toán bầy đàn có thể giúp tối ưu hóa điều khiển đèn giao thông và phân luồng giao thông để giảm thiểu tắc nghẽn. Chẳng hạn, PSO có thể được sử dụng để tìm ra các thiết lập đèn giao thông tối ưu, giúp giảm thời gian chờ đợi của các phương tiện và cải thiện lưu thông trên đường phố.

**f. Điều khiển robot tìm kiếm và cứu nạn**

**Ví dụ**: Trong các hoạt động cứu hộ, nhóm robot có thể được điều khiển bằng thuật toán bầy đàn để tìm kiếm và giải cứu nạn nhân trong các tình huống khẩn cấp như động đất hay cháy rừng. Các robot này sử dụng thuật toán như PSO hoặc ACO để khám phá môi trường, định vị nạn nhân và tạo ra lộ trình an toàn để đưa nạn nhân ra ngoài.

# **CHƯƠNG 3: THUẬT TOÁN TỐI ƯU BẦY ONG (ARTIFICIAL BEE COLONY)**

**1. Giới thiệu**

Thuật toán tối ưu hóa bằng tổ ong (Artificial Bee Colony), được viết tắt là ABC là một phương pháp tối ưu hóa lấy cảm hứng từ hành vi tìm kiếm thức ăn của bầy ong mật. Thuật toán này mô phỏng quá trình tìm kiếm và khai thác nguồn thức ăn của ba loại ong: ong thợ, ong quan sát và ong trinh sát. Được phát triển bởi Derviş Karaboğa vào năm 2005. Là thuật toán meta-heuristic để giải các bài toán gần đúng và các bài toán phức tạp. Theo đánh giá, ABC là thuật toán hiệu quả để giải quyết bài toán khó có nhiều điểm tối ưu cục bộ. Hiệu quả của thuật toán được đánh giá và kiểm tra trên các bài toán tối ưu hàm số thực. Kết quả thực nghiệm trên các hàm số thực cho ra kết quả nhanh và chính xác hơn.

Vào nằm 2005: Giáo sư Derviş Karaboğa giới thiệu thuật toán ABC lần đầu tiên. Ông và nhóm nghiên cứu đã phát triển thuật toán này dựa trên quan sát hành vi của ong thợ, ong quan sát và ong trinh sát trong quá trình tìm kiếm và khai thác nguồn thức ăn. Giai đoạn 2007-2010: Thuật toán ABC bắt đầu được áp dụng rộng rãi trong các lĩnh vực như tối ưu hóa hàm số, phân cụm dữ liệu, và các bài toán tối ưu hóa khác. Và từ 2010-nay: ABC tiếp tục được cải tiến và mở rộng ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm cả mạng nơron, xử lý ảnh, và các hệ thống phức tạp khác. Thuật toán ABC nổi bật nhờ khả năng tìm kiếm giải pháp hiệu quả và khả năng tránh được các điểm tối ưu cục bộ, giúp nó trở thành một công cụ mạnh mẽ trong lĩnh vực tối ưu hóa.

**2. Lý do ra đời**

Thuật toán ABC (Artificial Bee Colony) được phát triển dựa trên hành vi tìm kiếm thức ăn của đàn ong. Lý do ra đời của thuật toán này bao gồm:

- **Mô phỏng tự nhiên**: ABC được thiết kế để mô phỏng cách mà ong tìm kiếm và thu thập thức ăn, từ đó áp dụng vào các bài toán tối ưu hóa phức tạp.

**- Tối ưu hóa hiệu quả**: Thuật toán này nhằm cải thiện hiệu suất trong việc tìm kiếm giải pháp tối ưu cho các bài toán, đặc biệt là trong các lĩnh vực như tối ưu hóa hàm, lập lịch, và thiết kế.

**- Khả năng hội tụ nhanh**: ABC có khả năng hội tụ nhanh đến giải pháp tối ưu, nhờ vào việc khai thác và khám phá không gian tìm kiếm một cách hiệu quả.

**- Đơn giản và linh hoạt**: Thuật toán này dễ dàng áp dụng cho nhiều loại bài toán khác nhau và có thể điều chỉnh để phù hợp với các yêu cầu cụ thể.

**- Khả năng xử lý đa dạng**: ABC có thể xử lý các bài toán tối ưu hóa đa mục tiêu và các bài toán không có cấu trúc rõ ràng.

=> Tóm lại, thuật toán ABC ra đời nhằm mục đích cung cấp một phương pháp tối ưu hóa hiệu quả, dựa trên các nguyên tắc tự nhiên, với khả năng áp dụng rộng rãi trong nhiều lĩnh vực khác nhau.

**3. Lợi ích của thuật toán bầy ong**

Thuật toán ABC (Artificial Bee Colony) là một thuật toán tối ưu hóa dựa trên hành vi tìm kiếm thức ăn của ong. Dưới đây là một số lợi ích của thuật toán ABC:

- **Tính đơn giản**: Thuật toán ABC có cấu trúc đơn giản và dễ hiểu, giúp cho việc triển khai và điều chỉnh trở nên dễ dàng.

- **Khả năng tìm kiếm toàn cục**: ABC có khả năng tìm kiếm toàn cục tốt, giúp tránh bị mắc kẹt trong các cực tiểu cục bộ.

- **Tính linh hoạt**: Thuật toán có thể áp dụng cho nhiều loại bài toán tối ưu hóa khác nhau, từ tối ưu hóa hàm đến tối ưu hóa trong các bài toán phức tạp hơn.

- **Hiệu suất cao**: Trong nhiều trường hợp, ABC cho kết quả tối ưu tốt hơn so với các thuật toán tối ưu hóa khác, đặc biệt là trong các bài toán có không gian tìm kiếm lớn.

**- Khả năng mở rộng**: Thuật toán có thể dễ dàng mở rộng để xử lý các bài toán lớn hơn hoặc phức tạp hơn.

- **Không yêu cầu thông tin về gradient**: ABC không yêu cầu thông tin về gradient của hàm mục tiêu, điều này làm cho nó phù hợp với các bài toán không khả thi để tính toán gradient.

**4. Nguyên lý hoạt động và các bước thuật toán**

a, Nguyên lý hoạt động

Khởi tạo quần thể: Ban đầu, một quần thể các giải pháp ngẫu nhiên (tương ứng với các con ong) được tạo ra.

-Giai đoạn tìm kiếm thức ăn:

+Ong thợ: Các con ong thợ tìm kiếm nguồn thức ăn mới xung quanh các nguồn hiện có và đánh giá chất lượng của chúng.

+Ong quan sát: Các con ong quan sát chọn nguồn thức ăn dựa trên thông tin từ các con ong thợ và quyết định tìm kiếm xung quanh các nguồn thức ăn tốt nhất.

+Ong trinh sát: Nếu một nguồn thức ăn không được cải thiện sau một số lần lặp, con ong thợ sẽ trở thành ong trinh sát và tìm kiếm nguồn thức ăn mới ngẫu nhiên.

Cập nhật nguồn thức ăn: Các nguồn thức ăn được cập nhật dựa trên chất lượng của chúng, và quá trình này lặp lại cho đến khi đạt được điều kiện dừng (ví dụ: số lần lặp tối đa hoặc đạt được chất lượng mong muốn).

b, Các bước thuật toán

\*Sơ đồ thuật toán

A diagram of a flowchart

Description automatically generated

Sơ đồ chung của thuật toán bầy ong

Từ thuật toán trên ta có thể rút ra các bước giải quyết một bài toán ứng dụng với thuật toán bầy ong:

*Bước 1: Giai đoạn khởi tạo*

* Trong giai đoạn này, thuật toán tạo ra một **quần thể ban đầu** gồm các giải pháp ngẫu nhiên (vị trí các nguồn thức ăn trong không gian tìm kiếm).
* Mỗi giải pháp được đánh giá thông qua một **hàm mục tiêu**, nhằm xác định chất lượng của nó.

*Bước 2: Giai đoạn của ong thợ*

* Các ong thợ sẽ thực hiện tìm kiếm giải pháp mới dựa trên các giải pháp hiện tại.
* Mỗi ong thợ điều chỉnh giải pháp của mình bằng cách thay đổi vị trí (theo công thức toán học) trong không gian tìm kiếm, dựa trên thông tin từ các giải pháp khác.
* Nếu giải pháp mới tốt hơn, nó sẽ thay thế giải pháp hiện tại.

*Bước 3: Giai đoạn của ong quan sát*

* Ong **quan sát** sẽ lựa chọn các nguồn thức ăn (giải pháp) dựa trên **chất lượng** mà ong thợ đã báo cáo.
* Giải pháp tốt hơn sẽ có xác suất được chọn cao hơn.
* Sau đó, ong quan sát tiếp tục tìm kiếm giải pháp mới xung quanh các giải pháp được chọn, tương tự cách hoạt động của ong thợ.

*Bước 4: Giai đoạn của ong trinh sát*

* Sau giai đoạn của ong thợ và ong quan sát, thuật toán kiểm tra xem có nguồn thức ăn nào **không được cải thiện** trong một số lần lặp nhất định hay không.
* Nếu có, một **ong trinh sát** sẽ được kích hoạt để tìm kiếm một giải pháp hoàn toàn mới, thay thế giải pháp cũ không còn hiệu quả.

*Bước 5: Kiểm tra điều kiện dừng*

* Sau mỗi vòng lặp, thuật toán kiểm tra **điều kiện dừng**. Điều kiện này có thể là:

+ Số vòng lặp đạt giới hạn.

+ Đã tìm được giải pháp tối ưu đạt yêu cầu.

* Nếu điều kiện dừng **chưa thỏa mãn (False)**, thuật toán quay lại giai đoạn **ong thợ** và lặp lại quy trình.
* Nếu điều kiện dừng **thỏa mãn (True)**, thuật toán dừng và **báo cáo giải pháp tốt nhất** đã tìm được.

*Bước 6: Kết quả cuối cùng*

* Sau khi thuật toán hoàn tất, giải pháp tốt nhất (nguồn thức ăn có chất lượng cao nhất) được báo cáo. Đây là **giải pháp tối ưu** cho bài toán.

\*Công thức thuật toán:

**- Công thức cập nhật giải pháp (vị trí nguồn thức ăn)**

Khi một **ong thợ** hoặc **ong quan sát** tạo ra một giải pháp mới, công thức được sử dụng như sau:

= + ()

* : Giải pháp mới tại chiều j của ong i.
* : Vị trí hiện tại tại chiều j của ong i.
* Vị trí tại chiều j của một ong ngẫu nhiên k trong bầy (với k ≠ i).
* Một số ngẫu nhiên trong khoảng [−1,1], giúp điều chỉnh bước nhảy.

Mục tiêu của công thức này là di chuyển từ vị trí hiện tại đến một vị trí mới dựa trên sự khác biệt với một ong khác, qua đó khám phá không gian tìm kiếm.

**- Xác suất chọn nguồn thức ăn (giai đoạn ong quan sát)**

Ong quan sát chọn nguồn thức ăn dựa trên **xác suất tỷ lệ với chất lượng** của nguồn thức ăn. Công thức xác suất được tính như sau:

=

* Xác suất để chọn nguồn thức ăn iii.
* Độ phù hợp (fitness) của nguồn thức ăn iii.
* N: Tổng số nguồn thức ăn trong bầy.

Độ phù hợp thường được tính từ giá trị hàm mục tiêu ​ như sau:

Công thức này đảm bảo rằng các nguồn thức ăn tốt hơn (có giá trị hàm mục tiêu nhỏ hơn) sẽ có xác suất cao hơn được chọn.

**- Tìm kiếm giải pháp ngẫu nhiên (ong trinh sát)**

Nếu một nguồn thức ăn không được cải thiện sau một số vòng lặp nhất định (gọi là **giới hạn thử nghiệm**, thường được đặt trước), nó sẽ bị từ bỏ và một **ong trinh sát** được kích hoạt để tìm kiếm một giải pháp mới ngẫu nhiên:

+ rand(0,1).( )

* Vị trí mới tại chiều j của ong i.
* Giá trị giới hạn dưới tại chiều j.
* Giá trị giới hạn trên tại chiều j.
* rand (0,1)): Một số ngẫu nhiên trong khoảng từ 0 đến 1.

Công thức này đảm bảo rằng vị trí mới được chọn nằm trong phạm vi không gian tìm kiếm.

**- Điều kiện dừng**

Thuật toán sẽ dừng lại khi đạt một trong các điều kiện sau:

* Số vòng lặp tối đa được đặt trước.
* Không có cải thiện đáng kể trong giá trị hàm mục tiêu sau một số vòng lặp.
* Giải pháp đạt được thỏa mãn tiêu chí tối ưu hóa (ví dụ: giá trị hàm mục tiêu nhỏ hơn một ngưỡng xác định).

**Code Python:**

A computer screen shot of a code

Description automatically generated

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**A computer screen shot of a code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Chạy DEMO:**

* *Demo dạng 2D*

**A computer screen shot of a number

Description automatically generated**

* *Demo dạng 3D*

**A screenshot of a graph

Description automatically generated**

**5. Ứng dụng của thuật toán ABC**

**a. Tối ưu hóa lịch trình sản xuất**

**Ví dụ**: Trong ngành sản xuất, ABC có thể được sử dụng để tối ưu hóa lịch trình sản xuất nhằm giảm thiểu thời gian chờ đợi và tăng hiệu suất làm việc. Bằng cách tìm kiếm và điều chỉnh các lịch trình sản xuất khác nhau, ABC giúp các nhà máy sản xuất hoạt động hiệu quả hơn và giảm chi phí.

**b. Phân đoạn và nhận diện hình ảnh**

**Ví dụ**: Trong xử lý hình ảnh, ABC có thể được áp dụng để phân đoạn ảnh và nhận diện các đối tượng. Bằng cách tối ưu hóa các tham số của mô hình phân đoạn, ABC giúp cải thiện độ chính xác trong việc nhận diện và phân loại các đối tượng trong ảnh, ứng dụng trong y tế để phát hiện các khối u hoặc dị tật từ hình ảnh chụp X-quang hoặc MRI.

**c. Tối ưu hóa mạng nơ-ron**

**Ví dụ**: ABC được sử dụng để tối ưu hóa các tham số của mạng nơ-ron, giúp cải thiện hiệu suất học tập và dự đoán của mô hình. Chẳng hạn, trong nhận diện giọng nói, ABC giúp tối ưu hóa cấu trúc và tham số của mạng nơ-ron để nhận diện chính xác các lệnh từ người dùng.

**d. Quản lý năng lượng tái tạo**

**Ví dụ**: ABC có thể được áp dụng trong quản lý năng lượng tái tạo như tối ưu hóa việc sử dụng và lưu trữ năng lượng từ các nguồn năng lượng mặt trời và gió. Bằng cách tìm ra cách phân phối và lưu trữ năng lượng hiệu quả, ABC giúp giảm thiểu lãng phí năng lượng và tối đa hóa việc sử dụng các nguồn tài nguyên tái tạo.

**e. Dự đoán và phân tích tài chính**

**Ví dụ**: Trong lĩnh vực tài chính, ABC được sử dụng để dự đoán giá cổ phiếu và phân tích các mô hình tài chính phức tạp. Bằng cách tối ưu hóa các tham số của mô hình dự đoán, ABC giúp các nhà đầu tư đưa ra quyết định đúng đắn và giảm thiểu rủi ro.

**f. Quản lý và tối ưu hóa mạng lưới giao thông**

**Ví dụ**: ABC có thể được áp dụng để tối ưu hóa hệ thống giao thông, như điều khiển đèn giao thông và phân luồng giao thông để giảm thiểu tắc nghẽn. Thuật toán giúp xác định các chiến lược điều khiển tối ưu, cải thiện lưu lượng giao thông và giảm thời gian chờ đợi cho người tham gia giao thông.

**g. Thiết kế và tối ưu hóa mạng cảm biến không dây**

**Ví dụ**: ABC có thể được sử dụng để tối ưu hóa việc phân bố và cấu hình của các cảm biến trong mạng cảm biến không dây, giúp cải thiện hiệu suất và tiết kiệm năng lượng trong các ứng dụng giám sát môi trường, nông nghiệp thông minh, và quản lý đô thị.

# **CHƯƠNG 4: KẾT LUẬN**

Thuật toán tối ưu bầy đàn (PSO - Particle Swarm Optimization) và Thuật toán tối ưu bầy ong (Artificial Bee Colony) là hai thuật toán tối ưu hóa dựa trên hành vi của các loài động vật trong tự nhiên, mang lại nhiều ứng dụng thực tiễn trong các bài toán phức tạp.

Thuật toán bầy đàn là một lĩnh vực nghiên cứu trong khoa học máy tính và kỹ thuật lấy cảm hứng từ hành vi tập thể của các sinh vật như kiến, chim, cá và côn trùng. Những thuật toán này dựa trên cách các sinh vật trong bầy hoạt động cùng nhau để giải quyết các vấn đề phức tạp một cách hiệu quả và tối ưu.

Thuật toán bầy ong được đưa ra bởi Karaboga và đồng nghiệp đối với vấn đề tính toán tối ưu. Đây là một trong những thuật toán thuộc nhóm trí tuệ bầy đàn (Swarm Intelligence). Thuật toán mô phỏng cách thức tìm kiếm thức ăn của một bầy ong mật trong thực tế. Thuật toán rất đơn giản, và tối ưu ngẫu nhiên dựa vào quần thể đã chứng tỏ hiệu quả cao trong việc giải quyết nhiều bài toán tối ưu hóa phức tạp.

Mặc dù cả hai thuật toán đều có những ưu và nhược điểm riêng, nhưng chúng cho thấy tiềm năng to lớn trong việc giải quyết các bài toán tối ưu hóa thực tế. Việc áp dụng từng thuật toán phụ thuộc vào đặc điểm cụ thể của bài toán. PSO thường được sử dụng cho các bài toán tối ưu hóa đa chiều, liên tục và không tuyến tính, đặc biệt là những bài toán mà hàm mục tiêu không có dạng đạo hàm rõ ràng. ABC thường được áp dụng cho các bài toán tối ưu hóa rời rạc và liên tục, đặc biệt là những bài toán có không gian tìm kiếm lớn và phức tạp.

Cuối cùng, việc nghiên cứu và cải tiến hai thuật toán này hứa hẹn mở ra nhiều hướng đi mới trong lĩnh vực tối ưu hóa thông minh, góp phần giải quyết các bài toán khó khăn trong lĩnh vực khoa học và công nghệ.

# **Tài liệu tham khảo**

<https://www.studocu.com/vn/document/truong-dh-sai-gon-dai-hoc-chinh-quy/nhung-van-de-toan-cau/tieu-luan-ve-thuat-toan-bay-dan/109248962>

<https://en.wikipedia.org/wiki/Particle_swarm_optimization>

<https://www.tailieudaihoc.com/doc/255558.html>

<https://text.123docz.net/document/10552944-bao-cao-khoa-hoc-de-tai-toi-uu-hoa-bay-dan-pso.htm>

<https://scholar.dlu.edu.vn/thuvienso/bitstream/DLU123456789/207275/1/CVv146S72023334.pdf>