TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

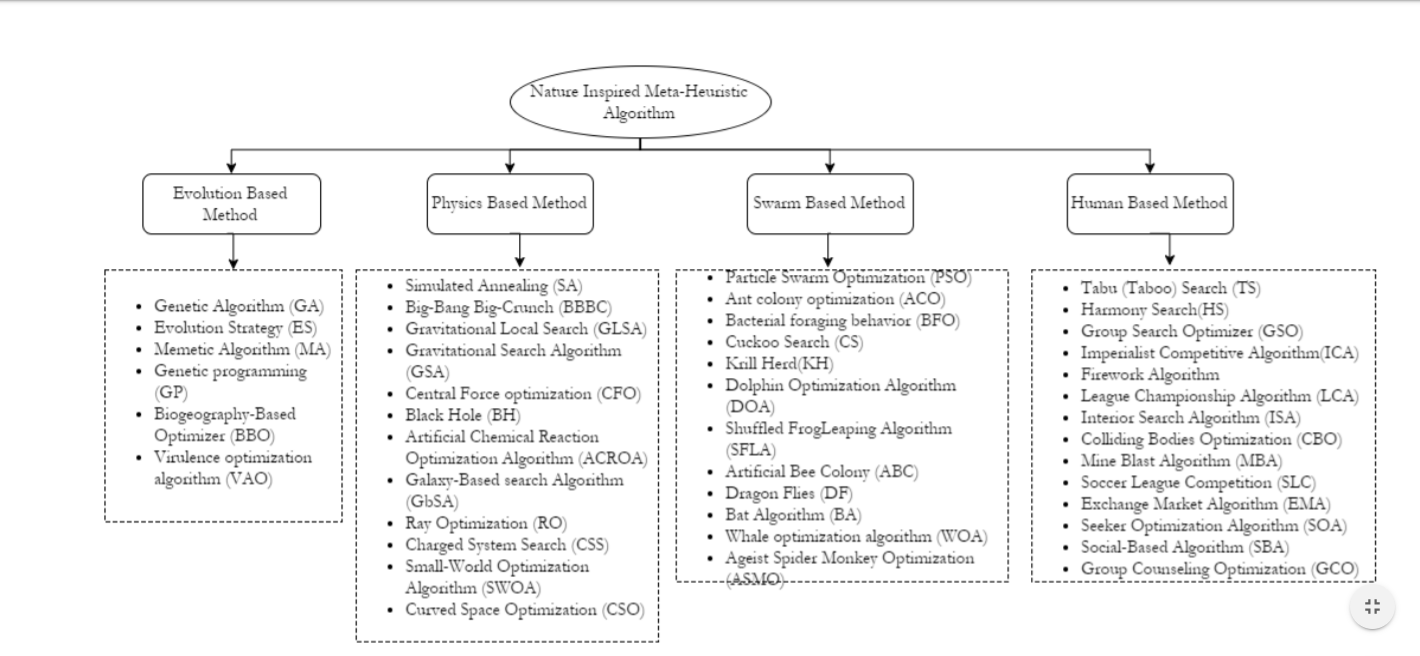
VIỆN CNTT&TT

**BÁO CÁO LAB**

**Ứng dụng thuật toán tìm kiếm bầy đàn(PSO) vào bài toán thực tế**

**Người thực hiện : Phùng Hà Dương.**

**1. Tổng quan thuật toán**

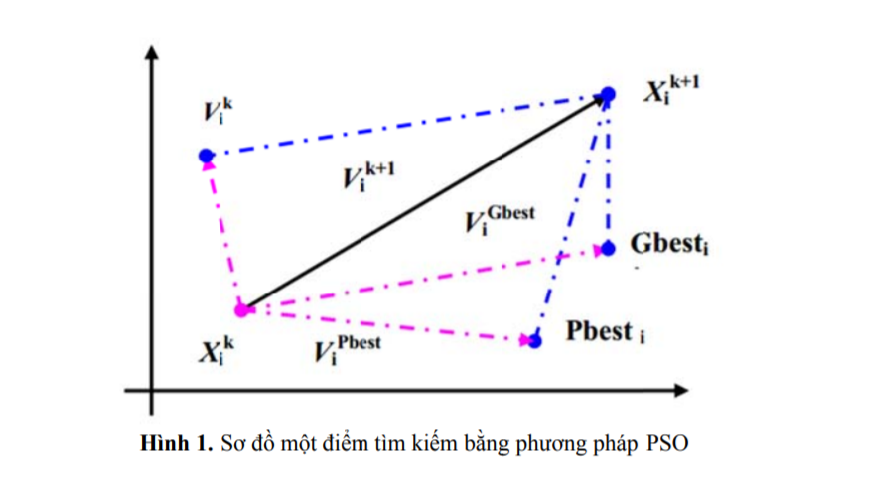


* Thuật toán tối ưu bầy đàn PSO là 1 trong số những thuật toán Swarm Based Method. Đây là một trong những thuật toán xây dựng dựa trên khái niệm trí tuệ bầy đàn để tìm ra lời giải tối ưu nhất. Nó mô hình hóa việc đàn chim đi tìm kiếm thức ăn.

Ví dụ: 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.

**2. Thuật toán**

PSO được khởi tạo bằng 1 nhóm cá thể ngẫu nhiên. Ta sẽ đi cập nhật những cá thể này qua các thế hệ. Cụ thể, trong mỗi thế hệ, mỗi cá thể được cập nhập theo 2 vị trí tốt nhất. Giá trị thứ nhất là vị trí tốt nhất mà nó từng đạt đến thời điểm hiện tại Pbest. Giá trị thứ hai là vị trí tốt nhất trong cả quá trình tìm kiếm cả quần thể từ trước đến hiện tại.



Trong đó:

Xki : vị trí cá thể thứ i tại thế hệ k

Vki : vận tốc cá thể thứ i tại thế hệ k

Xk+1i : vị trí cá thể thứ i tại thế hệ k+1

Vk+1i : vận tốc cá thể thứ i tại thế hệ k+1

Pbest : vị trí tốt nhất của quần thể i

Gbest : vị trí tốt nhất trong quần thể

Vận tốc và vị trí quần thể được tính:

Vk+1i  = w Vki  + c1 r1 (Pbest - Xki) + c2 r2 (Gbest - Xki) (1)

Xk+1i = Xki + Vk+1i (2)

W: hệ số quán tính

C1, C2 : các hệ số gia tốc

r1, r2 : các số ngẫu nhiên trong [0,1]

**CÁC BƯỚC THUẬT TOÁN PSO**

1. Khởi tạo quần thể:

(a) Thiết lập các hằng số: kmax, c1, c2.

(b) Khởi tạo ngẫu nhiên vị trí cá thể miền D.

(c) Khởi tạo ngẫu nhiên vận tốc cá thể.

(d) Đặt k = 1;

2. Tối ưu hóa:

(a) Đánh giá hàm fik bằng tọa độ của xki tính toán được trong không gian tìm kiếm.

(b) Nếu fik < fibest thì fibest = fik

(c) Nếu fik < fgbest thì fgbest = fik

(d) Nếu thỏa mãn tiêu chuẩn hội tụ thì dừng lại rồi thực hiện bước 3.

(e) Cập nhật tất cả các vận tốc vik và vị trí xik

(f) Tăng i. Nếu i > s thì đặt i = 1, tăng k

(g) Quay trở lại từ bước 2(a).

3. Kết thúc.

Với kmax là số lần lặp tối đa

**3. Áp dụng thuật toán**

Áp dụng vào tìm kiếm nhỏ nhất giá trị nhỏ nhất của hàm:

Fx2 = x2 + y2

Khởi tạo:

w = 0.5

c1 = 0.8

c2 = 0.9

n\_iterations = 50

n\_particles = 30

target\_error = 10-6

Khởi tạo 1 quần thể Particle có 30 cá thể nhận giá trị x nằm trong khoảng [-100, 100]

Giá trị vận tốc mỗi cá thể khởi tạo là [0, 0]

Vòng lặp sẽ lặp lại các bước sau:

* Cập nhật Pbest cho mỗi cá thể như sau: nếu fitness(particle) < Pbest -> update Pbest
* Cập nhật Gbest cho quần thể như sau: fitness( best\_fistness\_particle) < Gbest -> Update Gbest
* Di chuyển các cá thể:

+ Tính vận tốc mới từng cá thể theo công thức (1)

+ Cập nhật ví trí mới của từng cá thể đó theo công thức (2)

Kết quả thuật toán thu được:

