

ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ Thông tin và Truyền thông



SOICT

BÁO CÁO PROJECT

NHẬP MÔN KỸ THUẬT TRUYỀN THÔNG

Thuật toán Particle Swarm Optimization và các biến thể

Giảng viên hướng dẫn: TS. Trịnh Văn Chiến

Nhóm 24

Họ và Tên

Hà Trọng Thắng

Phạm Đức Anh

MSSV

20235827

20235652

Hà Nội, tháng 11 năm 2025

Mục lục

1	Tìm hiểu về thuật toán Particle Swarm Optimization (PSO)	3
1.1	Tổng quan về thuật toán	3
1.2	Nguyên lý hoạt động của thuật toán PSO	3
1.3	Các thành phần của thuật toán PSO	4
1.4	Quy trình thuật toán PSO	4
1.4.1	Khởi tạo quần thể	4
1.4.2	Đánh giá hàm mục tiêu	5
1.4.3	Cập nhật vị trí tối ưu của một cá thể nếu vị trí hiện tại của nó tốt hơn vị trí tốt nhất trước đó	5
1.4.4	Cập nhật vị trí tối ưu cục bộ lớn nhất của cá thể (P_{best}) và vị trí tối ưu toàn cục của cả đàn (G_{best})	5
1.4.5	Cập nhật vận tốc	5
1.4.6	Cập nhật vị trí	5
1.4.7	Điều kiện dừng	5
1.5	Mã giả	6
1.6	Độ phức tạp thuật toán	6
1.7	Tính hội tụ	7
2	Các biến thể của Thuật toán Particle Swarm Optimization	8
2.1	Hybrid PSO (PSO Lai ghép)	8
2.1.1	GA-PSO (Genetic Algorithm + PSO)	8
2.1.2	ESPO (Evolutionary PSO)	9
2.1.3	DEPSO (Differential Evolution + PSO)	10
2.2	Adaptive PSO (PSO Thích nghi)	11
2.3	PSO trong môi trường phức tạp	12
2.3.1	Multi-Objective PSO (MOPSO – Đa mục tiêu)	12
2.3.2	Dynamic PSO (Môi trường động)	12
2.4	Các biến thể khác	12
2.4.1	FIPS (Fully Informed Particle Swarm)	13
2.4.2	Bare – Bones (BBPSO – PSO “Trần trụi”)	13

Chương 1

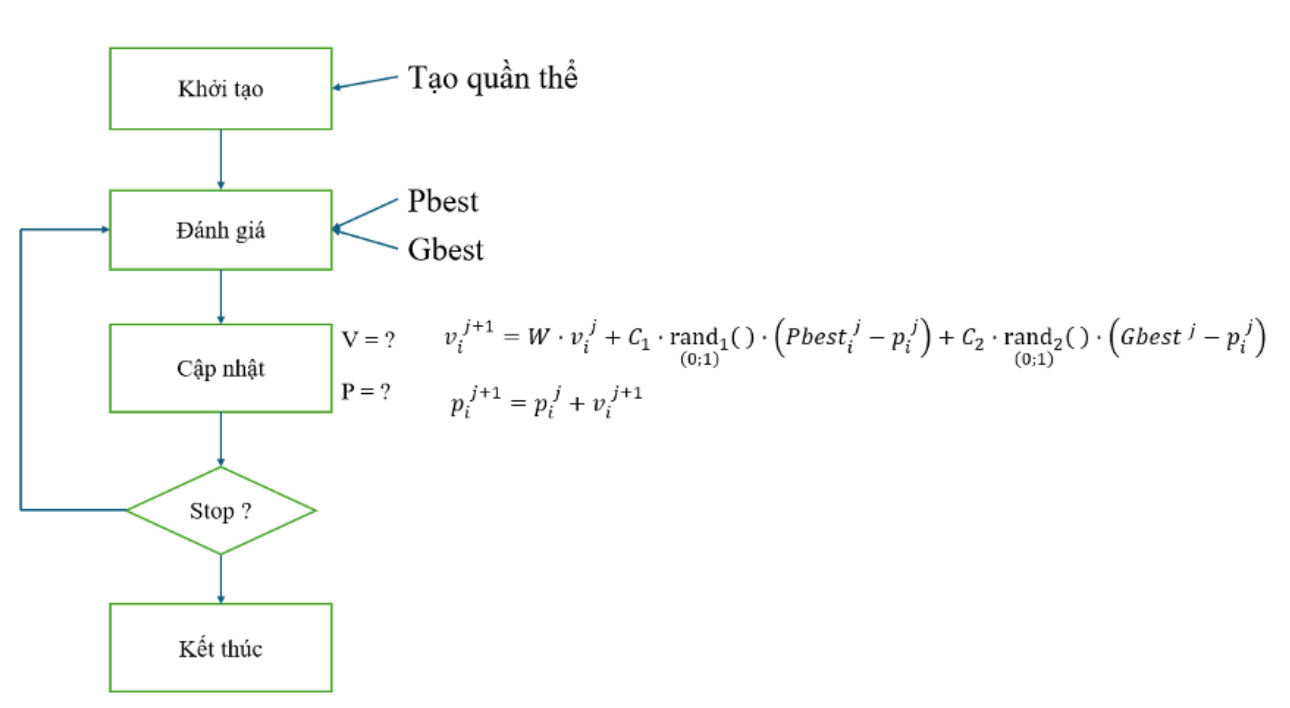
Tìm hiểu về thuật toán Particle Swarm Optimization (PSO)

1.1 Tổng quan về thuật toán

- PSO là thuật toán chung dựa trên kinh nghiệm của bầy đàn được đề xuất bởi Kennedy và Eberhat.
- Nó là khung thuật toán thông minh dựa trên bầy đàn, mô phỏng hành vi xã hội của bầy chim hoặc đàn cá khi đi tìm nguồn thức ăn.
- PSO sử dụng nhiều cá thể di chuyển xung quanh để tìm ra giải pháp tối ưu cục bộ trong không gian tìm kiếm. Mỗi cá thể sẽ ghi nhớ giải pháp tốt nhất của riêng mình và giải pháp tốt nhất từ các cá thể lân cận, sau đó đối chiếu với nhau để cập nhật lại giá trị cho giải pháp tối ưu cục bộ. Dựa vào đó để tìm ra giải pháp tối ưu nhất.

1.2 Nguyên lý hoạt động của thuật toán PSO

- Trong PSO, mỗi cá thể được gọi là hạt, đại diện cho một lời giải và di chuyển trong không gian tìm kiếm thông qua hai thông tin quan trọng vị trí và vận tốc.
- Sau khi cập nhật vận tốc, cá thể di chuyển đến vị trí mới và được đánh giá lại bằng hàm mục tiêu (dùng để đánh giá chất lượng của mỗi cá thể trong bầy).
- Dựa vào đó mà cập nhật vị trí tốt nhất mà chính nó từng đạt được (P_{best}) và vị trí tốt nhất mà toàn bầy đàn từng đạt được (G_{best}). Quá trình này được lặp lại cho đến khi đạt điều kiện dừng. Nhờ cơ chế vừa khai thác kinh nghiệm cá nhân vừa học hỏi từ tập thể, PSO có khả năng tìm ra lời giải tối ưu tốt một cách nhanh chóng và hiệu quả.



Hình 1.1: Sơ đồ mô tả nguyên lý hoạt động của thuật toán PSO

1.3 Các thành phần của thuật toán PSO

- **Swarm(bầy):** Tập các cá thể.
- **Particle(cá thể):** Ứng cử viên lời giải của bài toán.
 - **Vị trí:** p_i^j là vị trí của cá thể x ở vòng lặp i ở chiều j (với $j = 1, \dots, M$).
 - **Vận tốc:** v_i^j là vận tốc của cá thể x ở vòng lặp i ở chiều j (với $j = 1, \dots, M$).
 - **Vị trí tối ưu cục bộ lớn nhất của các cá thể:** $P_{best_x} = \max_{x=1..Y, j=1..M} \{p_i^j\}$
- **Vị trí tối ưu toàn cục cả đàn:** $G_{best} = \max_{i=1..Y} \{P_{best_i}\}$

1.4 Quy trình thuật toán PSO

1.4.1 Khởi tạo quần thể

- Tạo tập gồm Y cá thể, mỗi cá thể lặp N lần để tìm kiếm trong không gian M chiều.
- Với mỗi cá thể p_i , vị trí và vận tốc được khởi tạo ngẫu nhiên :

$$P_i^j(0) = (p_i^1, \dots, p_i^M)$$

$$V_i^j(0) = (v_i^1, \dots, v_i^M)$$

Trong đó:

- p_i^j là vị trí cá thể x ở vòng lặp i tại chiều j , ($j=1 \dots M$).
- v_i^j là vận tốc của cá thể x ở vòng lặp i tại chiều j , ($j=1 \dots M$).

1.4.2 Đánh giá hàm mục tiêu

- Với mỗi cá thể, tính giá trị $f(p_i^j)$.

1.4.3 Cập nhật vị trí tối ưu của một cá thể nếu vị trí hiện tại của nó tốt hơn vị trí tốt nhất trước đó

- Vị trí tối ưu cục bộ tại vị trí $j+1$ được cập nhật nếu giá trị của hàm mục tiêu tại vị trí $j+1$ lớn hơn hoặc bằng giá trị của hàm mục tiêu tại vị trí trước đó, ngược lại giá trị không thay đổi giữ lại vị trí của hàm mục tiêu trước đó.

$$pb_i^{j+1} = \begin{cases} p_i^{j+1}, & \text{nếu } f(p_i^{j+1}) \geq f(pb_i^j) \\ pb_i^j, & \text{ngược lại} \end{cases}$$

1.4.4 Cập nhật vị trí tối ưu cục bộ của cá thể và vị trí tối ưu toàn cục của cả đàn (P_{best} và G_{best})

- Sau khi mỗi cá thể tìm được vị trí tối ưu trên không gian, mỗi cá thể tìm được vị trí tối ưu nhất (P_{best}) bằng cách phân tích các vị trí tối ưu trên mỗi chiều và được xác định:

$$P_{best_x} = \max_{x=1..Y, j=1..M} \{pb_i^j\}$$

- Sau khi cập nhật vị trí tối ưu cục bộ P_{best} cho từng cá thể, ta xác định vị trí tối ưu toàn cục của cả quần thể (G_{best}). Đây là vị trí tốt nhất trong toàn bộ đàn tính đến vòng lặp hiện tại mỗi cá thể tìm được vị trí tối ưu trên không gian, mỗi cá thể tìm được vị trí tối ưu nhất (P_{best}) bằng cách phân tích các vị trí tối ưu trên mỗi chiều và được xác định:

$$G_{best} = \max_{i=1..Y} \{P_{best_i}\}$$

1.4.5 Cập nhật vận tốc

- Vận tốc của mỗi cá thể p_i tại vòng lặp $j+1$ được cập nhật theo công thức:

$$v_i^{j+1} = W \cdot v_i^j + C_1 \cdot rand_1() \cdot (P_{best_i}^j - p_i^j) + C_2 \cdot rand_2() \cdot (G_{best}^j - p_i^j)$$

Trong đó:

- W : Trọng số quán tính ($0 \rightarrow 1$).
- C_1 : Hệ số kinh nghiệm cá nhân.
- C_2 : Hệ số kinh nghiệm tập thể.

1.4.6 Cập nhật vị trí

- Sau khi tính được vận tốc mới, vị trí của cá thể được cập nhật theo:

$$p_i^{j+1} = p_i^j + v_i^{j+1}$$

1.4.7 Điều kiện dừng

- Thuật toán lặp lại từ bước 2 và sẽ kết thúc khi nào thỏa mãn điều kiện dừng, chẳng hạn như đạt đến số lần lặp tối đa hoặc đạt đến một giải pháp tối ưu đủ chính xác.

1.5 Mã giả

```
Khởi tạo quần thể gồm Y hạt với vị trí và vận tốc ngẫu nhiên
For mỗi hạt i:
    Tính giá trị  $f(p_i)$ 
     $Pbest_i = p_i$ 
Gbest = vị trí tốt nhất trong các Pbest
|
Lặp cho đến khi thỏa mãn điều kiện dừng:
    For mỗi hạt i:
        Cập nhật vận tốc:

$$v_i^{j+1} = W \cdot v_i^j + C_1 \cdot \text{rand}_1() \cdot (Pbest_i^j - p_i^j) + C_2 \cdot \text{rand}_2() (Gbest^j - p_i^j)$$

        Cập nhật vị trí:

$$p_i^{j+1} = p_i^j + v_i^{j+1}$$

        Tính  $f(p_i)$ 
        Nếu  $f(p_i)$  tốt hơn  $f(Pbest_i)$ :
             $Pbest_i = p_i$ 
        Cập nhật Gbest là Pbest tốt nhất
Kết thúc
```

Hình 1.2: Mã giả

1.6 Độ phức tạp thuật toán

Giả sử

- Y: Số hạt.
- M: Số chiều.
- T: Số vòng lặp.

PSO cần cập nhật vận tốc – vị trí – và đánh giá giá trị hàm mục tiêu cho từng hạt tại mỗi vòng lặp.

- Độ phức tạp thời gian: $O(T \times Y \times M)$
- Độ phức tạp bộ nhớ: $O(Y \times M)$

1.7 Tính hội tụ

PSO có xu hướng hội tụ nhanh nhờ cơ chế học hỏi hai hướng (cá nhân và tập thể). Để đảm bảo quỹ đạo hạt ổn định và hội tụ (không dao động vô hạn), các tham số phải thỏa mãn:

- **Trọng số quán tính (W):** $0 < W < 1$ để giữ sự ổn định quỹ đạo.
- **Hệ số (C_1, C_2):** $C_1 + C_2 < 4$ để đảm bảo hạt không dao động vô hạn.
- **Chiến lược W:** Có thể dùng công thức giảm dần tuyến tính theo thời gian để cân bằng:

$$w_t = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \cdot t$$

Chương 2

Các biến thể của Thuật toán Particle Swarm Optimization

2.1 Hybrid PSO (PSO Lai ghép)

Phương pháp "Mượn sức mạnh bên ngoài".

- **Vấn đề:** PSO cơ bản thường hội tụ sớm (Premature convergence) vào các cực trị địa phương và kém chính xác khi tinh chỉnh kết quả cuối cùng.
- **Giải pháp:** Kết hợp PSO với các thuật toán như Giải thuật di truyền (GA) hoặc Tìm kiếm cục bộ (Local Search) để cân bằng 2 yếu tố:
 - **Exploration:** Khả năng tìm kiếm diện rộng (thường mượn GA – Giải thuật di truyền).
 - **Exploitation:** Khả năng đào sâu vào một điểm để giá trị tốt nhất (thường mượn Local Search).

2.1.1 GA-PSO (Genetic Algorithm + PSO)

Đây là kết hợp giữa Trí tuệ bầy đàn (Swarm Intelligence) và Cơ chế chọn lọc tự nhiên (Natural Selection).

- **Ý tưởng chủ đạo:** Thay vì chỉ di chuyển, các hạt giờ đây có khả năng "**Sinh sản**" (**Breeding**) và "**Bị đào thải**".
- **Cơ chế hoạt động:**
 - **Selection Mechanism (Cơ chế chọn lọc):** Giúp các hạt nhảy từ vùng này sang vùng khác nhanh hơn.
 - **Replacement (Thay thế):** Những hạt có vị trí kém sẽ bị loại bỏ. Chúng sẽ thay thế bởi những bản sao của những hạt có vị trí tốt hơn.
- **Mục đích:** Tăng tốc độ hội tụ (accelerating convergence speed). Thay vì chờ cá thể kém hơn tìm hướng đi thì thay thế bằng cá thể tốt hơn.

2.1.2 ESPO (Evolutionary PSO)

Biến thể này tập trung vào việc “Đột biến các tham số” (Mutation of weights).

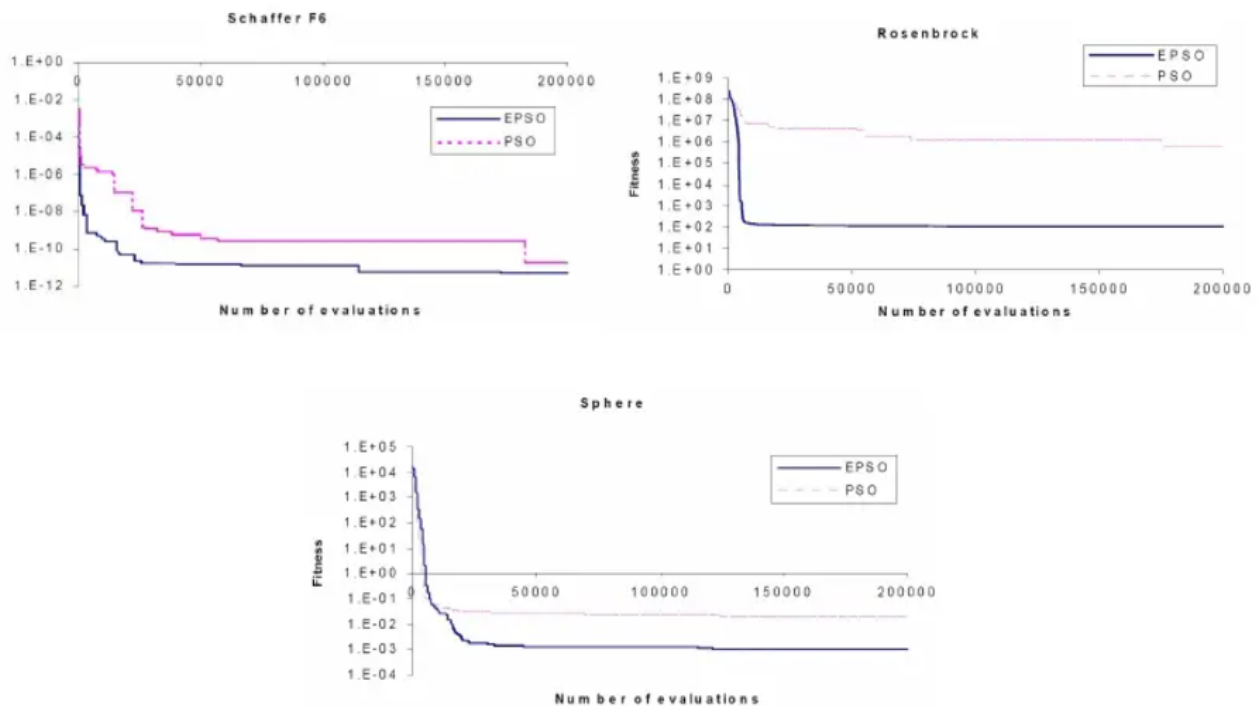
- **Sự khác biệt:** Trong PSO thường, các trọng số W (quán tính) thường cố định hoặc giảm dần đều. Còn trong ESPO các trọng số này bị nhiễu loạn ngẫu nhiên.
- Mỗi hạt sẽ mang một bộ tham số W riêng và tham số này thay đổi ngẫu nhiên sau mỗi vòng.

- **Công thức:**

$$w'_{ik} = w_{ik} + \tau \cdot N(0, 1)$$

(Nghĩa là: Quán tính mới = Quán tính cũ + Một chút nhiễu loạn ngẫu nhiên).

- Sau khi đột biến, hạt nào chạy tốt hơn với bộ tham số mới thì sẽ giữ được giữ lại (Survival Selection)
- **Mục đích:**
 - **Tự thích nghi (Self – Adapting):** Thuật toán tự tìm ra cách di chuyển tốt nhất.
 - **Thoát bế tắc:** Việc rung lắc các tham số giúp hạt không bị chạy theo lối mòn, dễ thoát ra khỏi các cực trị địa phương.
- **Hiệu quả:** Biểu đồ so sánh trên các hàm khó (Rosenbrock, Sphere) cho thấy đường ESPO (màu xanh đậm), tụt xuống rất nhanh so với PSO thường (màu hồng). Cho thấy được ESPO tìm nghiệm tối ưu tốt hơn hẳn.



Hình 2.1: Biểu đồ so sánh trên các hàm khó (Rosenbrock, Sphere)

2.1.3 DEPSO (Differential Evolution + PSO)

Đây có thể coi là một trong những biến thể mạnh, kết hợp với Tiến hóa vi phân (DE).

- **Điểm khác biệt so với PSO cơ bản:**

- **Cơ chế hoạt động:** PSO thường chạy liên tục công thức cập nhật vận tốc/vị trí. DEPSO chạy luân phiên với các nhịp (Nhịp lẻ để chạy PSO lao nhanh về đích, nhịp chẵn để chạy DE đột biến).
- **Đối tượng tác động:** PSO thường chỉ cập nhật vị trí hiện tại. DEPSO sử dụng toán tử DE để cập nhật Ký ức tốt nhất (P_i hay P_{best}).

- **Toán tử đột biến DE**

- Nó không tác động lên vị trí hiện tại, mà tác động lên P_{best} của hạt.
- Nó tạo ra một điểm thử nghiệm (Trial point T_i). Nếu điểm thử nghiệm này tốt hơn P_{best} cũ, nó sẽ thay thế P_{best} .
- Điều này giúp loại bỏ tình trạng “Hạt bị kẹt vào cực trị địa phương”.
- **Công thức:**

Trial point: $T_i = P_i$

For each dth dimension: if ($rand < CR$ or $d = k$) then $t_{id} = p_{gd} + \delta_{2d}$

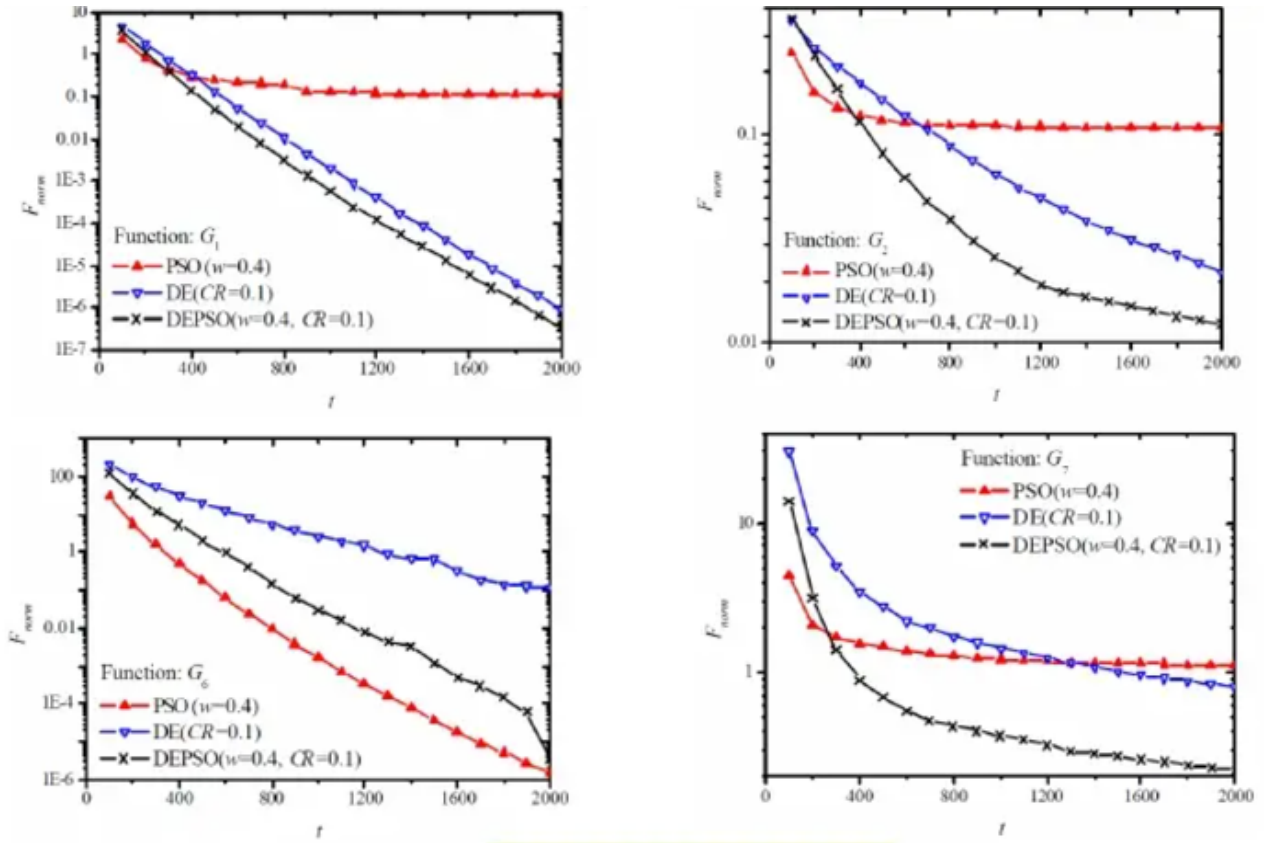
where k is a random integer value within $[1,n]$ which ensures the mutation in at least one dimension.

$$\delta_2 = \frac{1}{2}((P_{r1} - P_{r2}) + (P_{r3} - P_{r3}))$$

$$\text{if } fitness(T_i) > fitness(P_i) \text{ then } P_i \leftarrow T_i$$

- **Mục đích:**

- **Khắc phục điểm yếu chí mạng của PSO:** PSO thường bị phụ thuộc quá nhiều vào P_{best} . Nếu ngay từ đầu P_{best} bị sai thì hạt sẽ bị hút về đó mãi.
- **Đa dạng hóa hướng đi:** Toán tử DE giúp tạo ra các vị trí thử nghiệm khác biệt hoàn toàn so với đường đi hiện tại, giúp thuật toán nhảy cóc qua các vùng không gian mới mà PSO thường không với tới được.
- **Hiệu quả:** Nhìn vào biểu đồ, đường DEPSO (màu đen có dấu X) luôn nằm dưới cùng (thấp nhất = lỗi ít nhất = tốt nhất). Vượt trội hơn hẳn so với chạy PSO một mình (màu đỏ) hoặc DE một mình (màu xanh).



2.2 Adaptive PSO (PSO Thích nghi)

Đây là biến thể thông minh, "Biết thích nghi".

- **Vấn đề của PSO cơ bản:**

- Trong PSO chuẩn, các tham số như quán tính (W) và hệ số học tập ($C1, C2$) thường bị cài cứng.
- Vì vậy nên có Rủi ro: Nếu mới chạy được nửa thời gian mà đã tìm thấy vùng khả quan cao, nhưng vì W vẫn đang cao nên các hạt sẽ lướt qua mà không dừng lại để tìm hiểu.

- **Cơ chế hoạt động và thay đổi:** APSO sẽ liên tục theo dõi “Trạng thái tiến hóa” của đàn hạt. Nó đo khoảng cách trung bình giữa các hạt để biết đàn đang ở trạng thái nào.

- **Exploration (Khám phá):** Các hạt đang đứng xa nhau, cần W lớn để tìm kiếm rộng hơn.
- **Exploitation (Khai thác):** Các hạt đang tụ lại gần nhau, cần W nhỏ để tinh chỉnh.
- **Convergence (Hội tụ):** Đã tìm thấy đích.
- **Jumping out (Nhảy ra):** Đang bị kẹt, cần W tăng vọt để thoát ra.

- **Công thức điều chỉnh w (Adaptive Inertia Weight):** Thay vì giảm theo thời gian, w được tính theo hàm phản hồi (feedback):

$$w(t+1) = \frac{1}{1 + 1.5e^{-2.6f}} \in [0.4, 0.9]$$

Trong đó f là hệ số đánh giá độ phân tán của quần thể. Nếu phân tán rộng, f lớn thì W lớn và ngược lại.

$$w = w_{min} + (w_{max} - w_{min}) \frac{f(x) - f_{min}}{f_{avg} - f_{min}}$$

(Đây là một dạng phổ biến: w thay đổi dựa trên độ tốt $f(x)$ của hạt so với trung bình).

- **Tác dụng:**

- **Tối ưu hóa thời gian:** Không cần chờ hết giờ mới đi chậm lại. Nếu tìm thấy đích sớm, các hạt sẽ tự động dừng lại để tìm kiếm.
- **Tự giải cứu:** Nếu phát hiện ra các hạt đứng tùm lại một chỗ mà không tìm được gì tốt hơn (bị kẹt), cá thể sẽ tự động tăng ga để phá vỡ đội hình.

2.3 PSO trong môi trường phức tạp

Biến thể này xử lý các bài toán “không chuẩn mực” mà thực tế hay gặp.

2.3.1 Multi-Objective PSO (MOPSO – Đa mục tiêu)

- **Bài toán:** Tìm đích vừa **To nhất** vừa **Gần nhất**. Hai mục tiêu này thường mâu thuẫn (cái to thường xa). Không có một nghiệm duy nhất gọi là “Tốt nhất” (Global Best).
- **Thay đổi:**
 - Khái niệm Pareto Front: Thay vì tìm một điểm, MOPSO sẽ tìm một tập hợp các giải pháp (Ví dụ: Điểm A to nhưng xa, điểm B nhỏ nhưng gần). Không giải pháp nào hoàn toàn thắng giải pháp nào.
 - Kho lưu trữ (External Archive): Lưu lại tất cả các nghiệm Pareto tìm được.
 - Leader Selection (Chọn sếp): Khi cập nhật vận tốc, hạt sẽ không bay theo G_{best} duy nhất. Nó sẽ chọn một nghiệm trong “Kho lưu trữ” làm mục tiêu dẫn đường.
- **Công thức chọn G_{best} :** Dựa theo mật độ để đảm bảo các hạt tản ra tìm đủ các loại giải pháp.

2.3.2 Dynamic PSO (Môi trường động)

- **Bài toán:** Đích đến di chuyển.
- **Thay đổi:**
 - PSO thường sẽ hội tụ về vị trí cũ và đứng im đó.
 - Dynamic PSO sử dụng các hạt “Lính gác” (Sentries). Các hạt này đứng yên tại các vị trí G_{best} cũ và liên tục kiểm tra lại giá trị mục tiêu.
 - Nếu thấy giá trị mục tiêu thay đổi -> Kích hoạt báo động -> Tái khởi tạo một phần quần thể (Re-randomization) để đuổi theo mục tiêu mới.

2.4 Các biến thể khác

Biến thể này có thể thay đổi triết lý di chuyển.

2.4.1 FIPS (Fully Informed Particle Swarm)

- **Triết lý:** “Đừng theo ông sếp to nhất G_{best} , hãy nghe cả những người hàng xóm”.
- **Thay đổi:**
 - **PSO thường:** Vận tốc phụ thuộc vào bản thân P_{best} và lãnh đạo G_{best} .
 - **FIPS:** Vận tốc phụ thuộc vào tất cả hàng xóm xung quanh hạt đó.
- **Công thức cập nhật vận tốc:**

$$v_i^{t+1} = \chi \left[v_i^t + \frac{\sum_{k \in Neighbors} U(0,1) \cdot \varphi \cdot (p_k^t - x_i^t)}{|Neighbors|} \right]$$

(Hạt bị kéo bởi lực trung bình của tất cả hàng xóm)

- **Tác dụng:** Tránh bị một cá nhân xuất sắc “lừa” cả đội đi sai hướng. Thông tin đa chiều.

2.4.2 Bare – Bones (BBPSO – PSO “Trần trụi”)

Biến thể này bỏ qua khái niệm vận tốc.

- **Thay đổi:**
 - Loại bỏ hoàn toàn công thức vận tốc v .
 - Vị trí mới được chọn ngẫu nhiên và phân phối theo Gaussian (hình chuông) xung quanh trung điểm của G_{best} và P_{best} .
- **Công thức vị trí:**

$$x_{id}^{t+1} = N \left(\frac{p_{id} + g_d}{2}, |p_{id} - g_d| \right)$$

(Vị trí mới = Lấy mẫu từ phân phối chuẩn có trung bình là giữa 2 điểm tốt nhất).

- **Tác dụng:** Cực kì đơn giản, không cần chỉnh tham số quán tính hay vận tốc, gọn nhẹ cho code và hiệu quả bất ngờ