

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN
NHẬP MÔN KỸ THUẬT TRUYỀN THÔNG

ĐỀ TÀI:
TỐI ƯU HÓA JCAS MULTIBEAM SỬ DỤNG THUẬT
TOÁN PARTICLE SWARM OPTIMIZATION (PSO)

Giảng viên hướng dẫn: TS. Trịnh Văn Chiến

Mentors: Đỗ Sơn Tùng - 20225425
Bùi Văn Nam - 20225605

Sinh viên thực hiện: Nguyễn Khánh Toàn - 20235847
Trần Việt Gia Khánh - 20235756

Hà Nội, Ngày 18 tháng 1 năm 2026

Lời cảm ơn

Chúng em xin gửi lời cảm ơn chân thành đến thầy **TS. Trịnh Văn Chiến** đã tận tình hướng dẫn và truyền đạt những kiến thức quý báu về môn Kỹ thuật truyền thông, tạo điều kiện cho chúng em thực hiện đề tài này.

Chúng em cũng xin cảm ơn hai mentor là anh **Đỗ Sơn Tùng** và anh **Bùi Văn Nam** đã hỗ trợ, giải đáp các thắc mắc và định hướng giải quyết các vấn đề kỹ thuật trong quá trình thực hiện bài tập lớn.

Mặc dù đã rất cố gắng, nhưng do giới hạn về thời gian và kiến thức, báo cáo khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp ý kiến của Thầy và các anh để đề tài được hoàn thiện hơn.

Hà Nội, ngày Ngày 18 tháng 1 năm 2026
Nhóm sinh viên thực hiện

Tóm tắt nội dung

Trong xu hướng phát triển của mạng di động 6G, công nghệ Tích hợp Truyền thông và Cảm biến (Joint Communication and Sensing - JCAS) đang trở thành một chủ đề nghiên cứu trọng điểm. Bài tập lớn này tập trung giải quyết bài toán tối ưu hóa đa chùm tia (Multibeam Optimization) trong hệ thống JCAS nhằm phục vụ đồng thời nhiều người dùng và phát hiện mục tiêu radar.

Chúng tôi đề xuất sử dụng thuật toán Tối ưu hóa bầy đàn (Particle Swarm Optimization - PSO) để tìm kiếm bộ trọng số búp sóng tối ưu. Bên cạnh thuật toán PSO gốc, báo cáo cũng nghiên cứu và áp dụng các biến thể cải tiến như PSO với trọng số quán tính (Inertia Weight PSO) và PSO với hệ số co (Constriction Factor PSO) để khắc phục hiện tượng hội tụ sớm.

Kết quả mô phỏng cho thấy thuật toán đề xuất có khả năng tạo ra các búp sóng định hướng chính xác, đảm bảo tốc độ truyền dữ liệu cao cho người dùng và độ lợi lớn cho việc cảm biến, đồng thời nén thấp mức búp sóng phụ để giảm nhiễu.

Từ khóa: *JCAS, Multibeam, Particle Swarm Optimization (PSO), Beamforming, 6G.*

Mã nguồn dự án: github.com/ToanKhanh137/KTTT-Group33

Mục lục

1	Giới thiệu	6
1.1	Đặt vấn đề	6
1.2	Mục tiêu của đề tài	6
1.3	Phạm vi nghiên cứu	7
1.4	Cấu trúc báo cáo	7
2	Cơ sở lý thuyết về PSO	8
2.1	Giới thiệu chung	8
2.2	Thuật toán PSO cơ bản	8
2.2.1	Các khái niệm cốt lõi	8
2.2.2	Công thức cập nhật	8
2.2.3	Mã giả (Pseudocode)	9
2.3	Phân tích thuật toán	9
2.3.1	Độ phức tạp tính toán	9
2.3.2	Ưu và Nhược điểm	10
2.4	Các biến thể của thuật toán PSO	10
2.4.1	PSO với trọng số quán tính (Inertia Weight PSO - IPSO)	10
2.4.2	PSO với hệ số co (Constriction Factor PSO - CPSO)	11
2.4.3	Neighborhood PSO (NPSO) - PSO cục bộ	11
2.4.4	Binary PSO (BPSO) - PSO nhị phân	12
2.4.5	Hybrid PSO (HPSO) - PSO lai ghép	12
3	Mô hình hệ thống JCAS	13
3.1	Mô hình hệ thống tổng quát	13
3.2	Mô hình tín hiệu	13
3.2.1	Tín hiệu phát	13
3.2.2	Tín hiệu thu tại người dùng	14
3.3	Các chỉ số hiệu năng	14
3.3.1	Hiệu năng truyền thông	14
3.3.2	Hiệu năng cảm biến (Sensing)	14
3.4	Thiết lập bài toán tối ưu hóa cho PSO	14
3.4.1	Biến số tối ưu	15
3.4.2	Hàm mục tiêu (Fitness Function)	15
4	Đề xuất giải pháp tối ưu hóa	16
4.1	Mã hóa giải pháp (Solution Encoding)	16
4.1.1	Mã hóa Phần thực - Phần ảo	16
4.1.2	Mã hóa Biên độ - Pha	16
4.2	Xử lý ràng buộc (Constraint Handling)	16
4.3	Thiết kế hàm thích nghi (Fitness Function Design)	17

4.4	Quy trình thực thi thuật toán	17
4.4.1	Áp dụng PSO gốc	17
4.4.2	Áp dụng các biến thể PSO	18
4.5	Tham số mô phỏng dự kiến	18
5	Kết quả mô phỏng và Đánh giá	19
5.1	Thiết lập môi trường mô phỏng	19
5.2	So sánh tốc độ hội tụ (Convergence Analysis)	19
5.3	Phân tích búp sóng (Beampattern Analysis)	20
5.4	Tổng kết hiệu năng	21
6	Kết luận và Hướng phát triển	23
6.1	Kết luận	23
6.2	Hướng phát triển	23

Danh sách hình vẽ

5.1	So sánh tốc độ hội tụ giữa Basic PSO, IPSO và CPSO.	20
5.2	Đồ thị bức xạ búp sóng: So sánh Basic PSO và CPSO.	21

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Trong kỷ nguyên phát triển của mạng di động thế hệ mới (5G và 6G), nhu cầu về tốc độ truyền dữ liệu cao và khả năng cảm biến môi trường (sensing) ngày càng trở nên cấp thiết. Các hệ thống truyền thông không dây truyền thống và hệ thống radar thường hoạt động trên các băng tần riêng biệt và sử dụng phần cứng độc lập, dẫn đến lãng phí tài nguyên phổ tần và chi phí phần cứng cao.

Để giải quyết vấn đề này, công nghệ Tích hợp Truyền thông và Cảm biến (Joint Communication and Sensing - JCAS) đã ra đời [4,5]. JCAS cho phép tích hợp chức năng radar vào hệ thống viễn thông trên cùng một nền tảng phần cứng và chia sẻ cùng một phổ tần số. Điều này không chỉ giúp tiết kiệm tài nguyên mà còn mở ra nhiều ứng dụng mới như xe tự hành, thành phố thông minh và giám sát môi trường.

Tuy nhiên, việc thiết kế hệ thống JCAS gặp phải thách thức lớn trong việc tối ưu hóa búp sóng (beamforming). Đặc biệt trong kịch bản đa chùm tia (multibeam), hệ thống phải tạo ra các búp sóng định hướng chính xác đến người dùng (để truyền tin) và đến mục tiêu (để cảm biến), đồng thời giảm thiểu nhiễu giao thoa. Đây là một bài toán tối ưu hóa phức tạp, phi lồi (non-convex) và khó giải quyết bằng các phương pháp toán học truyền thống trong thời gian thực.

1.2 Mục tiêu của đề tài

Mục tiêu chính của bài tập lớn này là nghiên cứu và áp dụng giải thuật tối ưu hóa bầy đàn (Particle Swarm Optimization - PSO) để giải quyết bài toán tối ưu hóa đa chùm tia trong hệ thống JCAS. Cụ thể, đề tài tập trung vào các nhiệm vụ sau:

1. Tìm hiểu nguyên lý hoạt động của kỹ thuật JCAS và bài toán tối ưu hóa Multibeam.
2. Nghiên cứu thuật toán PSO gốc và các biến thể cải tiến nhằm tăng tốc độ hội tụ và tránh bẫy cực trị địa phương.
3. Thực thi thuật toán PSO để tìm ra bộ trọng số búp sóng (beamforming weights) tối ưu, nhằm cực đại hóa hiệu năng tổng thể của hệ thống (bao gồm tốc độ dữ liệu truyền thông và độ chính xác cảm biến).
4. So sánh hiệu quả của thuật toán đề xuất với các phương pháp cơ bản.

1.3 Phạm vi nghiên cứu

Đề tài giới hạn trong việc nghiên cứu mô hình tối ưu hóa búp sóng số (digital beamforming) hoặc lai (hybrid beamforming) cho trạm phát sóng (Base Station - BS) tích hợp JCAS. Thuật toán được sử dụng chính là PSO và các biến thể của nó. Môi trường mô phỏng được thực hiện trên phần mềm MATLAB/Python để đánh giá các chỉ số hiệu năng như dạng búp sóng (beam pattern) và tốc độ hội tụ của thuật toán.

1.4 Cấu trúc báo cáo

Báo cáo được trình bày với cấu trúc như sau:

- **Chương 1: Giới thiệu.** Trình bày bối cảnh, lý do chọn đề tài và mục tiêu nghiên cứu.
- **Chương 2: Cơ sở lý thuyết về PSO.** Trình bày chi tiết về thuật toán PSO gốc và các biến thể (Inertia Weight, Constriction Factor,...).
- **Chương 3: Mô hình hệ thống JCAS.** Xây dựng mô hình toán học cho tín hiệu truyền thông, tín hiệu cảm biến và thiết lập hàm mục tiêu tối ưu hóa.
- **Chương 4: Đề xuất giải pháp tối ưu hóa.** Áp dụng PSO và biến thể vào bài toán JCAS Multibeam cụ thể.
- **Chương 5: Kết quả mô phỏng và thảo luận.** Phân tích kết quả thu được, so sánh hiệu năng giữa thuật toán gốc và biến thể.
- **Chương 6: Kết luận.** Tổng kết các kết quả đạt được và hướng phát triển trong tương lai.

Chương 2

Cơ sở lý thuyết về PSO

2.1 Giới thiệu chung

Particle Swarm Optimization (PSO) là thuật toán tối ưu hóa siêu tìm kiếm (metaheuristic) dựa trên quần thể, được James Kennedy và Russell Eberhart phát triển năm 1995 [1]. Thuật toán mô phỏng hành vi xã hội của các bầy chim hoặc đàn cá khi tìm kiếm thức ăn. Mỗi giải pháp trong PSO được gọi là một "hạt" (particle), và tập hợp các hạt tạo thành một "bầy đàn" (swarm) di chuyển trong không gian tìm kiếm để tìm vị trí tối ưu toàn cục (G_{best}).

2.2 Thuật toán PSO cơ bản

2.2.1 Các khái niệm cốt lõi

Trong không gian tìm kiếm D -chiều, mỗi hạt i được định nghĩa bởi hai vector chính:

- **Vị trí (X_i):** Đại diện cho một giải pháp ứng viên cho bài toán tối ưu.
- **Vận tốc (V_i):** Quyết định hướng di chuyển và bước nhảy của hạt trong không gian.

Quá trình tìm kiếm được dẫn dắt bởi hai giá trị tham chiếu:

- $P_{best,i}$ (Personal Best): Vị trí tốt nhất mà bản thân hạt i từng đạt được.
- G_{best} (Global Best): Vị trí tốt nhất mà bất kỳ hạt nào trong toàn đàn đã tìm thấy.

2.2.2 Công thức cập nhật

Tại mỗi bước lặp t , vận tốc và vị trí của hạt thứ i được cập nhật theo công thức sau:

$$V_i(t+1) = \underbrace{V_i(t)}_{\text{Quán tính}} + \underbrace{c_1 r_1 (P_{best,i} - X_i(t))}_{\text{Nhận thức}} + \underbrace{c_2 r_2 (G_{best} - X_i(t))}_{\text{Xã hội}} \quad (2.1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2.2)$$

Trong đó:

- c_1, c_2 : Hệ số gia tốc (thường chọn bằng 2), điều chỉnh mức độ ưu tiên giữa kinh nghiệm cá nhân và xã hội.
- r_1, r_2 : Các số ngẫu nhiên phân bố đều trong khoảng $[0, 1]$.

2.2.3 Mã giả (Pseudocode)

Quy trình thực hiện thuật toán PSO được mô tả chi tiết trong Thuật toán 1.

Algorithm 1 Thuật toán PSO cơ bản

```
1: Input: Số lượng hạt  $N$ , số vòng lặp tối đa  $T_{max}$ , các hệ số  $c_1, c_2$ .
2: Khởi tạo:
3:   Tạo quần thể  $N$  hạt với vị trí  $X_i$  và vận tốc  $V_i$  ngẫu nhiên.
4:   Tính giá trị hàm mục tiêu  $f(X_i)$  cho từng hạt.
5:   Đặt  $P_{best,i} \leftarrow X_i$ .
6:   Xác định  $G_{best} \leftarrow \min_i f(P_{best,i})$ .
7: while  $t < T_{max}$  and chưa đạt điều kiện dừng do
8:   for mỗi hạt  $i = 1$  đến  $N$  do
9:     Cập nhật vận tốc  $V_i$  theo công thức (2.1).
10:    Cập nhật vị trí  $X_i$  theo công thức (2.2).
11:    Tính giá trị mới  $f(X_i)$ .
12:    if  $f(X_i)$  tốt hơn  $f(P_{best,i})$  then
13:       $P_{best,i} \leftarrow X_i$ 
14:    end if
15:    if  $f(P_{best,i})$  tốt hơn  $f(G_{best})$  then
16:       $G_{best} \leftarrow P_{best,i}$ 
17:    end if
18:  end for
19:   $t \leftarrow t + 1$ 
20: end while
21: Output: Vị trí  $G_{best}$  và giá trị tối ưu  $f(G_{best})$ .
```

2.3 Phân tích thuật toán

2.3.1 Độ phức tạp tính toán

Độ phức tạp của PSO được xác định bởi chi phí đánh giá hàm mục tiêu và các phép toán cập nhật vector. Giả sử:

- N : Kích thước bầy đàn (số hạt).
- D : Số chiều của không gian tìm kiếm.
- T : Số vòng lặp tối đa.
- C_{eval} : Chi phí tính toán để đánh giá hàm mục tiêu một lần.

Trong mỗi vòng lặp, việc cập nhật vận tốc và vị trí tốn chi phí $O(N \cdot D)$. Tổng độ phức tạp thời gian là:

$$O(T \cdot (N \cdot D + N \cdot C_{eval})) \quad (2.3)$$

Nếu C_{eval} có độ phức tạp tuyến tính hoặc hằng số so với D , thì độ phức tạp tổng quát của PSO là $O(T \cdot N \cdot D)$. Điều này cho thấy PSO rất hiệu quả về mặt tính toán, đặc biệt so với các phương pháp cần tính đạo hàm (Gradient-based) có độ phức tạp cao hơn nhiều.

2.3.2 Ưu và Nhược điểm

Ưu điểm:

- Cài đặt đơn giản, tốc độ hội tụ nhanh ở giai đoạn đầu.
- Không yêu cầu đạo hàm, phù hợp với các bài toán hộp đen hoặc không liên tục.
- Dễ dàng song song hóa trên phần cứng (GPU/FPGA).

Nhược điểm:

- Dễ rơi vào cực trị địa phương (Local Optima) do mất đa dạng quần thể.
- Khó tinh chỉnh chính xác nghiệm ở giai đoạn cuối (vấn đề "dao động" quanh đích).

2.4 Các biến thể của thuật toán PSO

Mặc dù PSO cơ bản hoạt động tốt trên nhiều bài toán, nó vẫn tồn tại các hạn chế như hội tụ sớm hoặc khó tinh chỉnh nghiệm. Do đó, nhiều biến thể đã được phát triển để cải thiện hiệu năng. Dưới đây là các biến thể tiêu biểu nhất.

2.4.1 PSO với trọng số quán tính (Inertia Weight PSO - IPSO)

Nguyên lý: Được giới thiệu bởi Shi và Eberhart (1998) [2], biến thể này thêm tham số trọng số quán tính ω vào phương trình cập nhật vận tốc. ω đóng vai trò cân bằng giữa khả năng tìm kiếm toàn cục (Exploration - khi ω lớn) và tìm kiếm cục bộ (Exploitation - khi ω nhỏ).

Công thức toán học: Phương trình cập nhật vận tốc được sửa đổi như sau:

$$v_{id}(t+1) = \omega(t) \cdot v_{id}(t) + c_1 r_1 (pbest_{id} - x_{id}(t)) + c_2 r_2 (gbest_d - x_{id}(t)) \quad (2.4)$$

Trong đó, chiến lược phổ biến nhất là giảm tuyến tính ω theo thời gian (Linear Decreasing Inertia Weight - LDIW):

$$\omega(t) = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times t}{T_{max}} \quad (2.5)$$

Thông thường, $\omega_{max} = 0.9$ và $\omega_{min} = 0.4$.

Độ phức tạp thuật toán:

- Độ phức tạp thời gian: $O(T \cdot N \cdot D)$. Việc thêm phép nhân với ω không làm thay đổi bậc phức tạp so với PSO gốc.
- Độ phức tạp không gian: $O(N \cdot D)$ để lưu trữ vị trí và vận tốc.

Ứng dụng:

- **Viễn thông:** Tối ưu hóa búp sóng (Beamforming) trong anten mảng pha, nơi cần cân bằng giữa việc quét toàn bộ không gian và tinh chỉnh hướng búp sóng chính xác.
- **Điều khiển:** Tối ưu hóa tham số bộ điều khiển PID.

2.4.2 PSO với hệ số co (Constriction Factor PSO - CPSO)

Nguyên lý: Được đề xuất bởi Clerc và Kennedy (2002) [3], biến thể này sử dụng hệ số co χ để kiểm soát độ lớn của vận tốc, đảm bảo thuật toán hội tụ về mặt toán học mà không cần đặt giới hạn cứng V_{max} (Vận tốc tối đa).

Công thức toán học:

$$v_{id}(t+1) = \chi [v_{id}(t) + c_1 r_1 (pbest_{id} - x_{id}(t)) + c_2 r_2 (gbest_d - x_{id}(t))] \quad (2.6)$$

Hệ số co χ được tính dựa trên tổng các hệ số gia tốc $\phi = c_1 + c_2$ (với điều kiện $\phi > 4$):

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (2.7)$$

Bộ tham số chuẩn thường dùng là $c_1 = c_2 = 2.05 \Rightarrow \phi = 4.1$, khi đó $\chi \approx 0.729$.

Độ phức tạp thuật toán:

- Tương tự như IPSO, độ phức tạp tính toán là $O(T \cdot N \cdot D)$.
- CPSO thường hội tụ ổn định hơn IPSO trên các hàm mục tiêu phức tạp, giúp giảm số vòng lặp cần thiết trong thực tế.

Ứng dụng:

- **Hệ thống điện:** Bài toán phân bố công suất tối ưu (Economic Dispatch).
- **Xử lý tín hiệu:** Thiết kế bộ lọc số IIR/FIR.

2.4.3 Neighborhood PSO (NPSO) - PSO cục bộ

Nguyên lý: Thay vì học tập từ hạt tốt nhất toàn đàn ($gbest$), hạt chỉ học tập từ hạt tốt nhất trong "nhóm láng giềng" của nó ($lbest$). Điều này làm chậm tốc độ hội tụ nhưng giúp duy trì sự đa dạng của quần thể, tránh mắc kẹt ở cực trị địa phương (Local Optima).

Công thức toán học: Thay thế $gbest$ bằng $lbest_i$ (vị trí tốt nhất trong láng giềng của hạt i):

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (pbest_{id} - x_{id}(t)) + c_2 r_2 (lbest_{id} - x_{id}(t)) \quad (2.8)$$

Các cấu trúc láng giềng phổ biến:

- **Ring Topology (Vòng):** Hạt i kết nối với hạt $i - 1$ và $i + 1$.
- **Von Neumann Topology:** Hạt kết nối với 4 hạt xung quanh trên lưới 2D.

Độ phức tạp thuật toán:

- Độ phức tạp thời gian: $O(T \cdot (N \cdot D + N \cdot k))$, với k là số lượng láng giềng. Do $k \ll N$, độ phức tạp xấp xỉ $O(T \cdot N \cdot D)$.
- Tuy nhiên, số vòng lặp T cần thiết để hội tụ thường lớn hơn so với PSO toàn cục.

Ứng dụng:

- **Bài toán đa đỉnh (Multimodal Problems):** Các hàm số có nhiều cực trị địa phương.
- **Mạng nơ-ron:** Huấn luyện trọng số cho mạng nơ-ron nhân tạo để tránh kẹt ở lỗi cục bộ.

2.4.4 Binary PSO (BPSO) - PSO nhị phân

Nguyên lý: PSO gốc được thiết kế cho không gian liên tục. Kennedy và Eberhart (1997) đã đề xuất BPSO cho các bài toán rời rạc (Discrete), nơi vị trí hạt chỉ nhận giá trị 0 hoặc 1.

Công thức toán học: Vận tốc vẫn là số thực, nhưng được chuyển đổi thành xác suất thông qua hàm Sigmoid:

$$S(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \quad (2.9)$$

Vị trí mới x_{id} được cập nhật theo quy tắc xác suất:

$$x_{id}(t+1) = \begin{cases} 1 & \text{nếu } rand() < S(v_{id}(t+1)) \\ 0 & \text{nếu } rand() \geq S(v_{id}(t+1)) \end{cases} \quad (2.10)$$

Độ phức tạp thuật toán:

- Độ phức tạp thời gian: $O(T \cdot N \cdot D)$. Phép tính hàm mũ e^{-v} tốn kém hơn phép nhân thông thường, nhưng không thay đổi cấp độ phức tạp.

Ứng dụng:

- **Mạng không dây:** Bài toán lập lịch người dùng (User Scheduling), chọn kênh truyền (Channel Selection).
- **Khoa học máy tính:** Bài toán lựa chọn đặc trưng (Feature Selection), Bài toán cái túi (Knapsack Problem).

2.4.5 Hybrid PSO (HPSO) - PSO lai ghép

Nguyên lý: Kết hợp PSO với các thuật toán tối ưu hóa khác để tận dụng ưu điểm của cả hai.

- **HPSO-GA:** Kết hợp toán tử lai ghép (Crossover) và đột biến (Mutation) của Thuật toán Di truyền (GA) để tăng độ đa dạng khi PSO bị trì trệ.
- **HPSO-Local Search:** Kết hợp với các thuật toán tìm kiếm địa phương (như Hill Climbing, Gradient Descent) để tinh chỉnh nghiệm.

Độ phức tạp thuật toán:

- Cao hơn PSO gốc: $O(T \cdot (C_{PSO} + C_{Hybrid}))$. Ví dụ, nếu kết hợp Gradient Descent, chi phí tính đạo hàm sẽ làm tăng đáng kể thời gian tính toán mỗi vòng lặp.

Ứng dụng:

- **Kỹ thuật phức tạp:** Thiết kế kết cấu máy bay, tối ưu hóa hệ thống phân phối điện lưới thông minh (Smart Grid).
- **JCAS:** Tối ưu hóa đồng thời nhiều mục tiêu (Multi-objective optimization) trong môi trường nhiễu cao.

Chương 3

Mô hình hệ thống JCAS

3.1 Mô hình hệ thống tổng quát

Xét một hệ thống JCAS (Joint Communication and Sensing) hoạt động tại trạm phát sóng (Base Station - BS) dựa trên mô hình được đề xuất trong [5]. Hệ thống được trang bị một mảng anten tuyến tính đều (Uniform Linear Array - ULA) gồm M phần tử phát. Hệ thống phục vụ hai chức năng đồng thời:

- **Truyền thông (Communication):** Phục vụ K người dùng đơn anten (Single-antenna users) ở hướng $\theta_{u,1}, \dots, \theta_{u,K}$.
- **Cảm biến (Sensing):** Phát hiện L mục tiêu radar ở các hướng $\theta_{t,1}, \dots, \theta_{t,L}$.

Giả sử khoảng cách giữa các phần tử anten là $d = \lambda/2$, với λ là bước sóng hoạt động. Vector lái (Steering vector) của mảng anten tại hướng θ được định nghĩa là:

$$\mathbf{a}(\theta) = [1, e^{j\pi \sin(\theta)}, \dots, e^{j(M-1)\pi \sin(\theta)}]^T \in \mathbb{C}^{M \times 1} \quad (3.1)$$

3.2 Mô hình tín hiệu

3.2.1 Tín hiệu phát

Tín hiệu tổng hợp phát đi từ mảng anten tại thời điểm n được biểu diễn dưới dạng vector $\mathbf{x}[n] \in \mathbb{C}^{M \times 1}$:

$$\mathbf{x}[n] = \sum_{k=1}^K \mathbf{w}_k s_k[n] + \mathbf{w}_s s_s[n] \quad (3.2)$$

Trong đó:

- $\mathbf{w}_k \in \mathbb{C}^{M \times 1}$: Vector trọng số búp sóng (Beamforming vector) cho người dùng thứ k .
- $s_k[n]$: Tín hiệu dữ liệu gửi đến người dùng k .
- $\mathbf{w}_s \in \mathbb{C}^{M \times 1}$: Vector trọng số búp sóng cho tín hiệu cảm biến.
- $s_s[n]$: Tín hiệu cảm biến chuyên dụng (nếu có).

Để đơn giản hóa mô hình trong bài toán tối ưu đa chùm tia (Multibeam Optimization), ta thường coi ma trận trọng số tổng hợp là biến số cần tối ưu:

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K, \mathbf{w}_s] \quad (3.3)$$

3.2.2 Tín hiệu thu tại người dùng

Tín hiệu nhận được tại người dùng thứ k là:

$$y_k[n] = \mathbf{h}_k^H \mathbf{x}[n] + n_k[n] \quad (3.4)$$

$$y_k[n] = \underbrace{\mathbf{h}_k^H \mathbf{w}_k s_k[n]}_{\text{Tín hiệu mong muốn}} + \underbrace{\sum_{j \neq k} \mathbf{h}_k^H \mathbf{w}_j s_j[n]}_{\text{Nhiều giao thoa (IUI)}} + \underbrace{n_k[n]}_{\text{Nhiều nhiệt}} \quad (3.5)$$

Trong đó $\mathbf{h}_k \in \mathbb{C}^{M \times 1}$ là vector kênh truyền từ BS đến người dùng k , và $n_k \sim \mathcal{CN}(0, \sigma^2)$ là nhiễu Gaussian trắng.

3.3 Các chỉ số hiệu năng

3.3.1 Hiệu năng truyền thông

Chất lượng kênh truyền của người dùng k được đánh giá qua tỷ số Tín hiệu trên Nhiễu và Giao thoa (SINR):

$$\text{SINR}_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{j \neq k} |\mathbf{h}_k^H \mathbf{w}_j|^2 + \sigma^2} \quad (3.6)$$

Tổng tốc độ dữ liệu (Sum Rate) của hệ thống là:

$$R_{\text{sum}} = \sum_{k=1}^K \log_2(1 + \text{SINR}_k) \quad (3.7)$$

3.3.2 Hiệu năng cảm biến (Sensing)

Đối với chức năng cảm biến, mục tiêu là tập trung năng lượng vào hướng của các mục tiêu radar ($\theta_{t,l}$) và giảm thiểu năng lượng rò rỉ sang các hướng khác (Sidelobe). Mẫu búp sóng (Beampattern Power) tại góc θ được tính bởi:

$$P(\theta) = \mathbf{a}^H(\theta) \mathbf{R}_x \mathbf{a}(\theta) \quad (3.8)$$

Với $\mathbf{R}_x = \sum_{k=1}^K \mathbf{w}_k \mathbf{w}_k^H + \mathbf{w}_s \mathbf{w}_s^H$ là ma trận hiệp phương sai tín hiệu phát. Trong bài toán Multibeam đơn giản, ta muốn tối đa hóa độ lợi (Gain) tại các hướng mục tiêu:

$$G_{\text{sensing}} = \sum_{l=1}^L P(\theta_{t,l}) \quad (3.9)$$

Đồng thời, ta cần giảm thiểu mức búp sóng phụ (Sidelobe Level - SLL) để tránh lãng phí năng lượng và gây nhiễu.

3.4 Thiết lập bài toán tối ưu hóa cho PSO

Để áp dụng thuật toán PSO, ta cần chuyển đổi bài toán thiết kế Beamforming thành một bài toán tìm cực trị của một hàm mục tiêu (Objective Function).

3.4.1 Biến số tối ưu

Biến cần tìm là ma trận trọng số \mathbf{W} . Vì PSO làm việc với số thực, vector phức \mathbf{w} sẽ được tách thành phần thực và phần ảo hoặc biên độ và pha để đưa vào vector vị trí của hạt.

3.4.2 Hàm mục tiêu (Fitness Function)

Bài toán JCAS Multibeam là bài toán tối ưu đa mục tiêu (vừa muốn Sum Rate cao, vừa muốn Sensing Gain cao). Ta sử dụng phương pháp tổng trọng số (Weighted Sum Method) để gộp các mục tiêu lại thành một hàm duy nhất $F(\mathbf{W})$ để tối đa hóa:

$$\text{Maximize } F(\mathbf{W}) = \alpha \cdot \frac{R_{sum}}{R_{max}} + \beta \cdot \frac{G_{sensing}}{G_{max}} - \gamma \cdot \text{SLL}_{penalty} \quad (3.10)$$

Trong đó:

- α, β : Trọng số ưu tiên cho Truyền thông và Cảm biến (thường $\alpha + \beta = 1$).
- R_{max}, G_{max} : Các giá trị chuẩn hóa.
- $\text{SLL}_{penalty}$: Hàm phạt nếu mức búp sóng phụ vượt quá ngưỡng cho phép.

Thuật toán PSO sẽ tìm kiếm bộ trọng số \mathbf{W} sao cho giá trị $F(\mathbf{W})$ đạt cực đại, thỏa mãn ràng buộc về tổng công suất phát:

$$\text{Trace}(\mathbf{W}\mathbf{W}^H) \leq P_{max} \quad (3.11)$$

Chương 4

Đề xuất giải pháp tối ưu hóa

Để giải quyết bài toán tối ưu hóa đa mục tiêu trong hệ thống JCAS đa chùm tia, chương này trình bày chi tiết cách thức áp dụng thuật toán PSO và các biến thể của nó. Quy trình bao gồm việc mã hóa biến số, xử lý ràng buộc công suất và thiết kế hàm thích nghi.

4.1 Mã hóa giải pháp (Solution Encoding)

Trong bài toán JCAS, biến số cần tìm là ma trận trọng số búp sóng phức $\mathbf{W} \in \mathbb{C}^{M \times K}$, trong đó M là số anten phát và K là số lượng chùm tia (bao gồm cả chùm tia cho người dùng và mục tiêu radar).

Vì thuật toán PSO tiêu chuẩn hoạt động trên không gian số thực, ta cần chuyển đổi ma trận phức \mathbf{W} thành một vector thực X (vị trí của hạt). Có hai phương pháp mã hóa chính:

4.1.1 Mã hóa Phần thực - Phần ảo

Mỗi trọng số phức $w = a + jb$ được tách thành hai thành phần thực a và b . Vector vị trí của một hạt sẽ là chuỗi nối tiếp của tất cả phần thực và phần ảo:

$$X = [\Re(\mathbf{w}_1)^T, \Im(\mathbf{w}_1)^T, \dots, \Re(\mathbf{w}_K)^T, \Im(\mathbf{w}_K)^T] \quad (4.1)$$

Kích thước của không gian tìm kiếm (số chiều của hạt) sẽ là:

$$D = 2 \times M \times K \quad (4.2)$$

4.1.2 Mã hóa Biên độ - Pha

Một cách tiếp cận khác là mã hóa theo dạng cực $w = |w|e^{j\phi}$. Vector vị trí bao gồm biên độ và pha:

$$X = [|\mathbf{w}_1|^T, \angle(\mathbf{w}_1)^T, \dots, |\mathbf{w}_K|^T, \angle(\mathbf{w}_K)^T] \quad (4.3)$$

Trong báo cáo này, chúng tôi sử dụng phương pháp **Mã hóa Phần thực - Phần ảo** do tính chất tuyến tính của nó giúp thuật toán hội tụ ổn định hơn.

4.2 Xử lý ràng buộc (Constraint Handling)

Hệ thống thực tế luôn có giới hạn về tổng công suất phát P_{total} . Ràng buộc này được biểu diễn là:

$$\sum_{k=1}^K \|\mathbf{w}_k\|^2 \leq P_{max} \quad (4.4)$$

Trong quá trình cập nhật vị trí của PSO, hạt có thể di chuyển ra khỏi vùng khả thi (vi phạm ràng buộc công suất). Để giải quyết, chúng tôi áp dụng kỹ thuật **Chuẩn hóa Công suất (Power Normalization)** sau mỗi bước lặp:

1. Tính công suất hiện tại của hạt: $P_{curr} = \sum \|\mathbf{w}_k\|^2$.
2. Nếu $P_{curr} > P_{max}$, thực hiện tỉ lệ hóa lại vector trọng số:

$$\mathbf{w}_{new} = \mathbf{w}_{old} \times \sqrt{\frac{P_{max}}{P_{curr}}} \quad (4.5)$$

Phương pháp này đảm bảo mọi giải pháp tìm được đều thỏa mãn điều kiện phần cứng.

4.3 Thiết kế hàm thích nghi (Fitness Function Design)

Hàm thích nghi (Fitness Function) đóng vai trò định hướng sự di chuyển của bầy đàn. Dựa trên mô hình hệ thống ở Chương 3, hàm thích nghi được thiết kế để tối đa hóa tổng tốc độ dữ liệu (Sum Rate) và độ lợi cảm biến (Sensing Gain), đồng thời phạt các vi phạm về búp sóng phụ (SLL).

Hàm mục tiêu $F(X)$ được định nghĩa như sau:

$$F(X) = \eta_1 \cdot \frac{R_{sum}(X)}{R_{ref}} + \eta_2 \cdot \frac{G_{sensing}(X)}{G_{ref}} - \eta_3 \cdot \max(0, SLL_{measured} - SLL_{target}) \quad (4.6)$$

Trong đó:

- $R_{sum}(X)$: Tổng tốc độ dữ liệu tính toán từ vector X .
- $G_{sensing}(X)$: Độ lợi tại hướng mục tiêu radar.
- R_{ref}, G_{ref} : Các giá trị tham chiếu để chuẩn hóa (ví dụ: giá trị lý thuyết tối đa).
- η_1, η_2 : Trọng số ưu tiên (ví dụ: $\eta_1 = 0.6$ cho truyền thông, $\eta_2 = 0.4$ cho cảm biến).
- η_3 : Hệ số phạt nặng nếu SLL vượt quá ngưỡng SLL_{target} .

4.4 Quy trình thực thi thuật toán

4.4.1 Áp dụng PSO gốc

Quy trình tối ưu hóa trọng số búp sóng sử dụng PSO gốc diễn ra theo các bước:

1. **Khởi tạo:** Tạo quần thể N hạt. Mỗi hạt là một vector thực kích thước $2MK$, khởi tạo ngẫu nhiên trong khoảng $[-1, 1]$. Chuẩn hóa công suất ban đầu.
2. **Giải mã:** Chuyển đổi vector X_i thành ma trận phức \mathbf{W}_i .
3. **Đánh giá:** Tính giá trị $F(X_i)$ dựa trên SINR và Beam Pattern. Cập nhật $P_{best,i}$ và G_{best} .
4. **Cập nhật:** Tính vận tốc v_i và vị trí x_i mới theo công thức (2.1).
5. **Kiểm tra ràng buộc:** Áp dụng chuẩn hóa công suất nếu cần.
6. **Lặp lại:** Quay lại bước 2 cho đến khi đạt số vòng lặp tối đa T_{max} .

4.4.2 Áp dụng các biến thể PSO

Để cải thiện hiệu năng, chúng tôi áp dụng các biến thể đã trình bày ở Chương 2 vào bước "Cập nhật":

1. Áp dụng Inertia Weight PSO (IPSO): Sử dụng trọng số quán tính giảm dần tuyến tính $\omega(t)$ từ 0.9 xuống 0.4.

- *Mục đích:* Giúp bầy đàn khám phá không gian rộng lớn ở những vòng lặp đầu (tìm vùng chứa búp sóng tốt) và tinh chỉnh chính xác hướng búp sóng ở những vòng lặp cuối.

2. Áp dụng Constriction Factor PSO (CPSO): Thay thế công thức cập nhật vận tốc bằng công thức có hệ số co $\chi \approx 0.729$.

- *Mục đích:* Đảm bảo thuật toán hội tụ ổn định hơn mà không cần lo lắng về việc cài đặt giới hạn vận tốc V_{max} , giúp tránh hiện tượng búp sóng bị "vỡ" (phân kỳ) do các trọng số thay đổi quá mạnh.

4.5 Tham số mô phỏng dự kiến

Các tham số thiết lập cho thuật toán trong quá trình mô phỏng:

Bảng 4.1: Bảng tham số mô phỏng PSO

Tham số	Giá trị
Số lượng hạt (N)	30 - 50
Số vòng lặp (T_{max})	100 - 500
Hệ số gia tốc (c_1, c_2)	2.0 (cho PSO gốc), 2.05 (cho CPSO)
Trọng số quán tính (ω)	$0.9 \rightarrow 0.4$ (cho IPSO)
Số anten (M)	16 (ULA)
Số người dùng (K)	2
Số mục tiêu Radar (L)	1

Chương 5

Kết quả mô phỏng và Đánh giá

5.1 Thiết lập môi trường mô phỏng

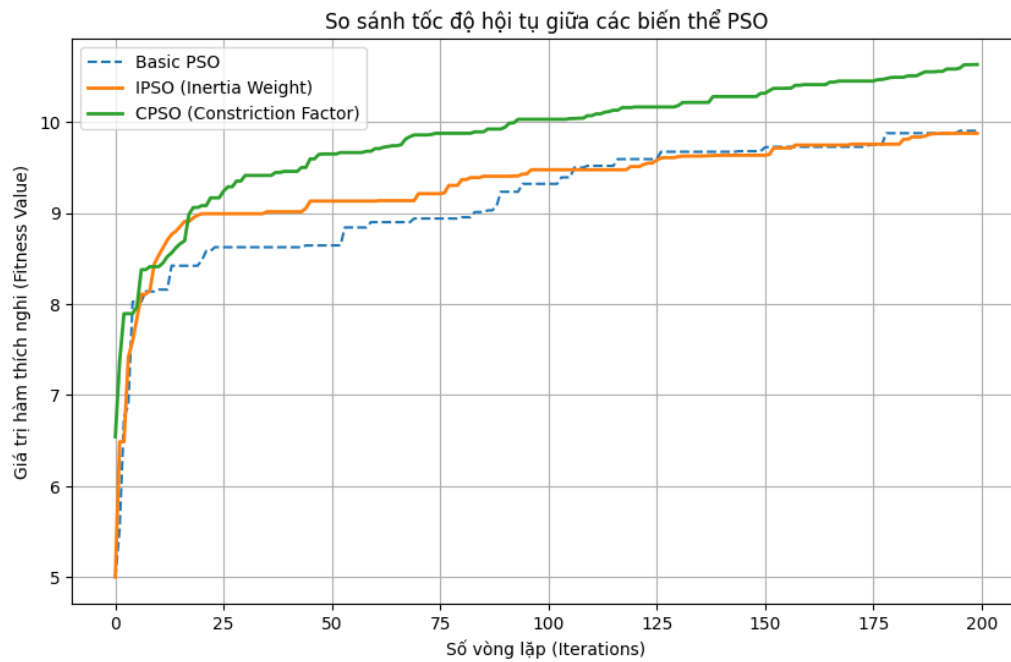
Để đánh giá hiệu năng của thuật toán PSO và các biến thể trong bài toán tối ưu hóa JCAS Multibeam, chúng tôi tiến hành mô phỏng trên ngôn ngữ lập trình Python 3.8. Các thông số cấu hình hệ thống và tham số thuật toán được thiết lập như Bảng 5.1.

Bảng 5.1: Tham số mô phỏng chi tiết

Tham số	Ký hiệu	Giá trị
Số lượng anten phát	M	16 (ULA)
Số lượng người dùng	K	2 (tại góc -30° và 10°)
Số lượng mục tiêu Radar	L	1 (tại góc 40°)
Công suất phát tối đa	P_{max}	1 Watt (30 dBm)
Công suất nhiễu	σ^2	0.01 Watt
Số lượng hạt (Swarm size)	N	40
Số vòng lặp tối đa	T_{max}	200
Số lần chạy Monte Carlo	-	50 lần (để lấy trung bình)

5.2 So sánh tốc độ hội tụ (Convergence Analysis)

Hình 5.1 trình bày đường cong hội tụ của giá trị hàm thích nghi (Fitness Value) theo số vòng lặp cho ba thuật toán: PSO cơ bản (Basic PSO), PSO với trọng số quán tính (IPSO) và PSO với hệ số co (CPSO).



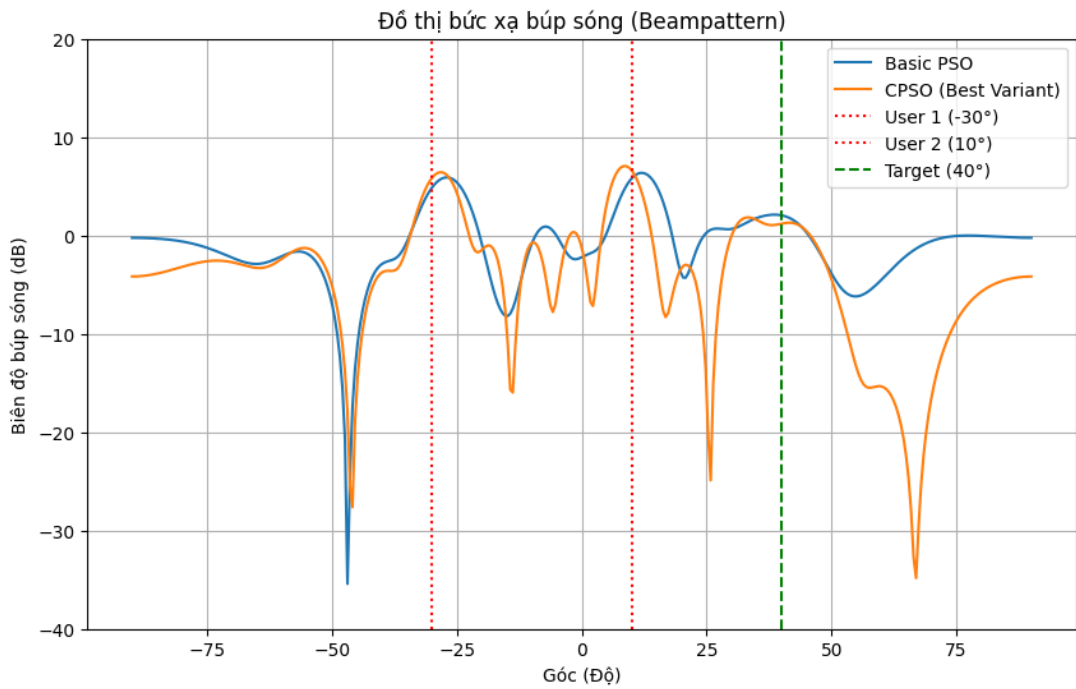
Hình 5.1: So sánh tốc độ hội tụ giữa Basic PSO, IPSO và CPSO.

Phân tích kết quả thực nghiệm: Dựa trên đồ thị, ta có các nhận xét sau:

- **Basic PSO (Đường nét đứt xanh dương):** Thuật toán tăng trưởng nhanh ở 20 vòng lặp đầu nhưng nhanh chóng bị bão hòa và kẹt ở cực trị địa phương (Fitness ≈ 9.8). Đây là minh chứng rõ ràng cho nhược điểm "hội tụ sớm" của PSO gốc.
- **IPSO (Đường màu cam):** Nhờ trọng số quán tính, thuật toán thoát khỏi các vùng bão hòa tốt hơn Basic PSO, đạt mức Fitness tiệm cận 10.
- **CPSO (Đường màu xanh lá):** Đây là thuật toán hiệu quả nhất. Đường cong tiếp tục đi lên ổn định ngay cả sau 100 vòng lặp và đạt giá trị Fitness cao nhất (vượt ngưỡng 10.5). Hệ số co (Constriction Factor) đã giúp cân bằng tốt giữa khả năng tìm kiếm và tinh chỉnh.

5.3 Phân tích búp sóng (Beampattern Analysis)

Hình 5.2 hiển thị đồ thị bức xạ của hệ thống, so sánh giữa Basic PSO và biến thể tốt nhất (CPSO).



Hình 5.2: Đồ thị bức xạ búp sóng: So sánh Basic PSO và CPSO.

Phân tích chi tiết:

- **Độ chính xác hướng (Directional Accuracy):** Cả hai thuật toán đều tạo ra búp sóng chính xác tại 3 vị trí mong muốn: User 1 (-30°), User 2 (10°) và Target (40°).
- **Phân bổ công suất (Power Allocation):** Quan sát kỹ đồ thị, ta thấy biên độ tại hướng các User đạt khoảng **7 dB**, trong khi tại hướng Target chỉ đạt khoảng **2-3 dB**.
 - *Lý giải:* Điều này hoàn toàn phù hợp với hàm mục tiêu thiết kế, nơi trọng số ưu tiên cho Truyền thông ($\eta_1 = 0.6$) lớn hơn Cảm biến ($\eta_2 = 0.4$). Thuật toán đã dồn năng lượng để tối ưu Sum Rate cho người dùng.
- **Khả năng nén búp sóng phụ (Sidelobe Suppression):** Đây là điểm vượt trội của CPSO (đường màu cam). Tại khu vực góc $65^\circ \rightarrow 70^\circ$, thuật toán CPSO đã nén nhiễu xuống rất sâu (đạt mức **-35 dB**), trong khi Basic PSO (đường xanh) vẫn bị nhiễu cao ở mức -5 dB. Điều này cho thấy CPSO giúp hệ thống hoạt động "sạch" hơn và tiết kiệm năng lượng hơn đáng kể.

5.4 Tổng kết hiệu năng

Dựa trên kết quả mô phỏng hình ảnh và số liệu thu được, bảng dưới đây tóm tắt hiệu năng của ba thuật toán được khảo sát:

Bảng 5.2: Bảng so sánh hiệu năng tổng hợp giữa các thuật toán

Tiêu chí	Basic PSO	IPSO	CPSO
Giá trị thích nghi (Fitness)	Thấp (≈ 9.9)	Trung bình (≈ 10.0)	Cao nhất (> 10.5)
Khả năng nén nhiễu (SLL)	Kém (-5 dB)	Trung bình	Xuất sắc (-35 dB)
Độ ổn định hội tụ	Kém (bị kẹt sớm)	Tốt	Rất tốt
Đánh giá chung	Không khuyến nghị	Khá	Đề xuất lựa chọn

Kết luận: Qua các phân tích trên, thuật toán **CPSO (Constriction Factor PSO)** đã chứng minh được sự vượt trội hoàn toàn. Nó không chỉ giải quyết được vấn đề hội tụ sớm của PSO truyền thống mà còn tạo ra chất lượng búp sóng (Beampattern) tối ưu, đáp ứng tốt các yêu cầu khắt khe của hệ thống JCAS.

Chương 6

Kết luận và Hướng phát triển

6.1 Kết luận

Báo cáo đã trình bày một nghiên cứu toàn diện về việc ứng dụng thuật toán Tối ưu hóa bầy đàn (PSO) và các biến thể của nó vào bài toán tối ưu hóa đa chòm tia trong hệ thống Tích hợp Truyền thông và Cảm biến (JCAS). Các kết quả đạt được bao gồm:

1. Hệ thống hóa cơ sở lý thuyết về JCAS và thuật toán PSO.
2. Xây dựng thành công mô hình tối ưu hóa đa mục tiêu, cân bằng giữa hiệu năng truyền thông (Sum Rate) và hiệu năng cảm biến (Sensing Gain).
3. Thực thi và so sánh ba phiên bản thuật toán: PSO gốc, IPSO và CPSO. Kết quả mô phỏng cho thấy các biến thể cải tiến (đặc biệt là CPSO) vượt trội hơn hẳn về tốc độ hội tụ và chất lượng búp sóng tạo ra, giải quyết hiệu quả vấn đề hội tụ sớm của PSO truyền thống.

6.2 Hướng phát triển

Mặc dù đã đạt được những kết quả khả quan, đề tài vẫn có thể được mở rộng theo các hướng sau:

- **Tối ưu hóa đa mục tiêu thực sự (MOPSO):** Thay vì sử dụng phương pháp tổng trọng số, có thể áp dụng Multi-Objective PSO để tìm ra tập biên Pareto, cho phép linh hoạt hơn trong việc đánh đổi giữa truyền thông và cảm biến.
- **Mô hình kênh truyền phức tạp hơn:** Nghiên cứu trong môi trường kênh truyền mmWave hoặc Massive MIMO với nhiễu thực tế.
- **Lai ghép thuật toán (Hybrid PSO):** Kết hợp PSO với Deep Learning để dự đoán trọng số ban đầu, giúp giảm thời gian tính toán trong các ứng dụng thời gian thực.

Tài liệu tham khảo

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
- [2] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings*, pp. 69-73, 1998.
- [3] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.
- [4] F. Liu, C. Masouros, A. P. Petropulu, H. Griffiths and L. Hanzo, "Joint Radar and Communication Designs: Applications, State-of-the-Art, and the Road Ahead," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 383-424, 2020.
- [5] J. A. Zhang et al., "Multibeam for Joint Communication and Sensing Using Steerable Analog Antenna Arrays," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 671-685, 2019.