

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



## **BÁO CÁO**

### **THUẬT TOÁN BẦY ONG ABC, CÁC BIẾN THỂ VÀ ÁP DỤNG TRONG BÀI TOÁN TỐI ƯU**

**Tên môn học:** Nhập môn Kỹ thuật truyền thông

**Mã môn học:** IT4593

**Nhóm:** 91

**Mentor:** Nguyễn Quang Đông  
Nguyễn Minh Quân

<b>Thành viên</b>	<b>MSSV</b>
Đào Thái Hoàng	20235720
Trần Thu Phương	20235811

Ngày 18 tháng 1 năm 2025

## MỤC LỤC

<b>CHƯƠNG 1. THUẬT TOÁN BẦY ONG NHÂN TẠO (ABC) .....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mô tả bài toán .....	1
1.3 Cơ sở lý thuyết thuật toán ABC.....	2
1.3.1 Ý tưởng sinh học.....	2
1.3.2 Ánh xạ sang bài toán tối ưu.....	2
1.4 Mô hình hóa bài toán .....	2
1.4.1 Công thức tính độ thích nghi (Fitness).....	2
1.4.2 Cơ chế tìm kiếm lân cận (Ong thợ và Ong quan sát) .....	3
1.4.3 Cơ chế chọn lọc (Ong quan sát) .....	3
1.4.4 Cơ chế sinh ngẫu nhiên (Ong trinh sát).....	4
1.5 Mã giả thuật toán.....	5
1.6 Đánh giá Ưu và Nhược điểm.....	6
1.6.1 Ưu điểm .....	6
1.6.2 Nhược điểm.....	6
<b>CHƯƠNG 2. CÁC BIẾN THỂ THUẬT TOÁN ABC .....</b>	<b>7</b>
2.1 Giới thiệu chung về các biến thể ABC .....	7
2.2 Phân tích Biến thể GABC .....	7
2.2.1 Hạn chế của Thuật toán ABC Chuẩn.....	7
2.2.2 Cơ chế hoạt động của GABC .....	7
2.3 Phân tích Biến thể qABC (Quick ABC) .....	8
2.3.1 Ý tưởng cốt lõi.....	8
2.3.2 Phương trình tìm kiếm qABC.....	8

2.3.3 Công thức Hybrid GABC - qABC .....	8
2.3.4 So sánh trực quan.....	9
2.4 So sánh Tổng hợp các Biến thể.....	9
<b>CHƯƠNG 3. TÌM HIỂU BÀI TOÁN TỐI ƯU TRONG TRUYỀN THÔNG ....</b>	<b>10</b>
3.1 Giới thiệu.....	10
3.1.1 Bối cảnh.....	10
3.1.2 Động lực và mục tiêu nghiên cứu.....	11
3.2 Phát biểu bài toán tối ưu .....	12
3.2.1 Mô hình hệ thống và Thông số .....	12
3.2.2 Định nghĩa các tập, biến, tham số .....	12
3.2.3 Biến cần tìm và ý nghĩa .....	13
3.3 Hàm mục tiêu.....	14
3.3.1 Hàm mục tiêu $f_1$ : Thiết kế búp sóng theo GLS.....	14
3.3.2 Hàm mục tiêu $f_2$ : Tối ưu hóa Vector Pha (ILS) .....	14
3.3.3 Hàm mục tiêu $f_3$ : Tối ưu hóa Pha Kết hợp ( $\phi$ ) .....	14
3.3.4 Hàm mục tiêu $f_4$ : Tối ưu hóa Lượng tử hóa (IGSS-Q).....	15
3.4 Ràng buộc tối ưu .....	15
3.4.1 Ràng buộc tổng công suất phát.....	15
3.4.2 Ràng buộc phân bố năng lượng .....	15
3.4.3 Ràng buộc phần cứng tương tự .....	15
3.4.4 Ràng buộc về thời gian/độ trễ.....	15
3.5 Ví dụ minh họa phép tính .....	15
3.5.1 Cấu hình giả lập.....	15
3.5.2 Quy trình tính toán.....	16

3.6 Thuật toán metaheuristic áp dụng.....	17
3.6.1 Mô tả thuật toán.....	17
3.6.2 Các tham số thuật toán .....	18
3.7 Kết quả và so sánh .....	18
3.7.1 Testcase 1: Hệ thống nhỏ ( $M = 2, K = 4$ ).....	18
3.7.2 Testcase 2: Hệ thống trung bình ( $M = 4, K = 10$ ).....	19
3.8 Kết luận .....	20
<b>Tài liệu tham khảo.....</b>	<b>20</b>

# CHƯƠNG 1. THUẬT TOÁN BẦY ONG NHÂN TẠO (ABC)

## 1.1 Đặt vấn đề

Trong những năm gần đây, các thuật toán tối ưu hóa dựa trên trí tuệ bầy đàn (Swarm Intelligence) như Genetic Algorithm (GA), Particle Swarm Optimization (PSO) hay Ant Colony Optimization (ACO) đã chứng minh được hiệu quả trong việc giải quyết các bài toán phức tạp.

Thuật toán Bầy ong nhân tạo **ABC (Artificial Bee Colony)**, được đề xuất bởi D.T. Phạm và các cộng sự (2005), là một trong những kỹ thuật meta-heuristic nổi bật. Thuật toán này mô phỏng hành vi tìm kiếm thức ăn thông minh của đàn ong mật. ABC kết hợp giữa khả năng tìm kiếm ngẫu nhiên (của ong trinh sát) và tìm kiếm cục bộ (của ong thợ và ong quan sát) để đưa ra lời giải tối ưu.

Báo cáo này sẽ trình bày cơ sở lý thuyết, mô hình toán học, mã giả của thuật toán ABC và phân tích các biến thể nâng cao.

## 1.2 Mô tả bài toán

Bài toán tối ưu hóa là quá trình tìm kiếm giá trị tốt nhất (cực đại hoặc cực tiểu) cho một hàm mục tiêu  $f(x)$  trong một không gian tìm kiếm xác định. Trong ngữ cảnh của thuật toán ABC, bài toán thường được đặt ra như sau:

Cho một hàm mục tiêu  $f(x)$  cần tối ưu hóa (ví dụ: tìm giá trị nhỏ nhất), với vector biến  $x = (x_1, x_2, \dots, x_D)$  thuộc không gian  $D$ -chiều. Mỗi biến  $x_j$  bị giới hạn trong khoảng  $[x_{min,j}, x_{max,j}]$ .

Mục tiêu là tìm vector  $x^*$  sao cho  $f(x^*)$  đạt giá trị tối ưu toàn cục (global optimum). Bài toán này thường gặp nhiều thách thức như:

- Không gian tìm kiếm rộng lớn và đa chiều.
- Hàm mục tiêu có thể không liên tục, không khả vi hoặc có nhiều cực trị cục bộ (local optima), dễ khiến các thuật toán truyền thống bị mắc kẹt.
- Yêu cầu về thời gian tính toán và độ chính xác cao.

Thuật toán ABC giải quyết bài toán này bằng cách mô phỏng quá trình tìm kiếm nguồn thức ăn của bầy ong, trong đó vị trí của một nguồn thức ăn đại diện cho một nghiệm tiềm năng  $x$  và lượng mật hoa của nguồn đó tương ứng với chất lượng (fitness) của nghiệm.

### 1.3 Cơ sở lý thuyết thuật toán ABC

#### 1.3.1 Ý tưởng sinh học

Trong tự nhiên, một đàn ong tối ưu hóa quá trình tìm mật thông qua việc phân chia nhiệm vụ:

- **Ong thợ (Employed Bees):** Tìm kiếm nguồn thức ăn, ghi nhớ vị trí và lượng mật, sau đó quay về tổ và thực hiện “vũ điệu” để chia sẻ thông tin.
- **Ong quan sát (Onlooker Bees):** Đứng chờ ở tổ, quan sát điệu nhảy của ong thợ và lựa chọn nguồn thức ăn hứa hẹn nhất để khai thác.
- **Ong trinh sát (Scout Bees):** Bay ngẫu nhiên để tìm các nguồn thức ăn mới khi nguồn cũ đã cạn kiệt.

#### 1.3.2 Ánh xạ sang bài toán tối ưu

Thuật toán ABC chuyển đổi các khái niệm sinh học thành các thành phần toán học:

- **Nguồn thức ăn:** Tương ứng với một **nghiệm** (lời giải) khả thi của bài toán.
- **Lượng mật hoa:** Tương ứng với giá trị **hàm thích nghi (fitness)** của nghiệm (chất lượng lời giải).
- **Vị trí nguồn mật:** Tương ứng với các tham số biến quyết định trong không gian tìm kiếm  $D$ -chiều.

### 1.4 Mô hình hóa bài toán

Để áp dụng thuật toán ABC vào giải quyết bài toán tối ưu, chúng ta cần xây dựng một mô hình toán học cụ thể, bao gồm các công thức tính toán độ thích nghi, cơ chế sinh nghiệm mới và quy tắc chọn lọc.

Giả sử không gian tìm kiếm có số chiều là  $D$ , kích thước quần thể (số nguồn thức ăn) là  $SN$  (thường bằng  $n/2$  tổng số ong).

#### 1.4.1 Công thức tính độ thích nghi (Fitness)

Giá trị hàm mục tiêu  $f(x)$  của bài toán được chuyển đổi sang giá trị thích nghi để đánh giá chất lượng nghiệm. Theo tài liệu tham khảo, công thức tính như sau:

$$fitness_i = \begin{cases} \frac{1}{1 + f_i} & \text{nếu } f_i \geq 0 \\ 1 + |f_i| & \text{nếu } f_i < 0 \end{cases}$$

Trong đó  $f_i$  là giá trị hàm mục tiêu của nghiệm thứ  $i$ .

### 1.4.2 Cơ chế tìm kiếm lân cận (Ong thợ và Ong quan sát)

Để tạo ra một nghiệm ứng viên mới  $v_{ij}$  từ nghiệm hiện tại  $x_{ij}$ , thuật toán sử dụng công thức biến đổi:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Trong đó:

- $k \in \{1, \dots, SN\}$  và  $k \neq i$  là một chỉ số ngẫu nhiên khác.
- $j \in \{1, \dots, D\}$  là chỉ số chiều được chọn ngẫu nhiên.
- $\phi_{ij}$  là một số ngẫu nhiên trong khoảng  $[-1, 1]$ .

Sau khi sinh ra  $v_i$ , thuật toán áp dụng cơ chế **tham lam (Greedy Selection)**: nếu  $fitness(v_i) > fitness(x_i)$  thì cập nhật  $x_i = v_i$ , ngược lại giữ nguyên  $x_i$  và tăng biến đếm số lần không cải thiện ( $trial_i$ ).

### 1.4.3 Cơ chế chọn lọc (Ong quan sát)

Ong quan sát chọn nguồn thức ăn dựa trên cơ chế **Vòng quay Roulette (Roulette Wheel Selection)**. Thay vì chỉ tính xác suất đơn thuần, thuật toán thực hiện việc cộng dồn giá trị thích nghi để chọn ra nghiệm tốt nhất một cách ngẫu nhiên nhưng có trọng số.

Quy trình chọn lọc được thực hiện như sau:

1. Tính tổng độ thích nghi của toàn bộ quần thể:

$$TotalFit = \sum_{n=1}^{SN} fitness_n$$

2. Sinh một số ngẫu nhiên  $r$  trong khoảng  $[0, TotalFit]$ .
3. Duyệt qua các nguồn thức ăn và cộng dồn giá trị thích nghi vào biến tích lũy  $acc$ .  
Nguồn thức ăn thứ  $i$  được chọn là nguồn đầu tiên thỏa mãn điều kiện:

$$\sum_{k=1}^i fitness_k \geq r$$

Cơ chế này đảm bảo các nguồn thức ăn có độ thích nghi ( $fitness$ ) càng cao thì đoạn giá trị trên “bánh xe” càng lớn, do đó xác suất được chọn bởi  $r$  sẽ cao hơn.

### 1.4.4 Cơ chế sinh ngẫu nhiên (Ong trình sát)

Nếu một nghiệm  $x_i$  không được cải thiện sau một số lần lặp vượt quá giới hạn ( $trial_i > limit$ ), nghiệm đó bị coi là “cạn kiệt”. Ong thợ tương ứng trở thành ong trình sát và sinh ra một nghiệm mới ngẫu nhiên:

$$x_{ij} = x_{min,j} + rand(0, 1) \times (x_{max,j} - x_{min,j})$$



## 1.5 Mã giả thuật toán

---

### Algorithm 1 Thuật toán ABC (Artificial Bee Colony) với Roulette Wheel

---

**Require:**  $SN$  (số nguồn thức ăn),  $limit$ ,  $MaxCycle$ ,  $D$  (số chiều)

**Ensure:**  $GlobalBest$  (Nghệ thuật tốt nhất)

```

1: Khởi tạo: Sinh ngẫu nhiên  $SN$  nguồn thức ăn  $x_i$ .
2: Tính  $f(x_i)$  và  $fitness(x_i)$  cho mọi  $i$ .
3:  $trial_i \leftarrow 0$  cho mọi  $i$ ;  $cycle \leftarrow 1$ 
4: while  $cycle \leq MaxCycle$  do
5:   for  $i = 1$  to  $SN$  do
6:     Sinh nghiệm mới  $v_i$  từ  $x_i$  (tìm kiếm lân cận).
7:     Đánh giá  $fitness(v_i)$ .
8:     if  $fitness(v_i) > fitness(x_i)$  then
9:        $x_i \leftarrow v_i$ ;  $trial_i \leftarrow 0$ 
10:    else
11:       $trial_i \leftarrow trial_i + 1$ 
12:    end if
13:  end for
14:   $TotalFit \leftarrow \sum fitness$ 
15:  for  $t = 1$  to  $SN$  do
16:     $r \leftarrow rand(0, TotalFit)$ 
17:     $acc \leftarrow 0$ 
18:     $SelectedInd \leftarrow 0$ 
19:    for  $idx = 1$  to  $SN$  do ▷ Chọn nguồn theo bánh xe Roulette
20:       $acc \leftarrow acc + fitness_{idx}$ 
21:      if  $acc \geq r$  then
22:         $SelectedInd \leftarrow idx$ 
23:        break
24:      end if
25:    end for
26:     $i \leftarrow SelectedInd$ 
27:    Sinh nghiệm mới  $v_i$  từ  $x_i$ .
28:    Đánh giá và chọn tham lam giữa  $v_i$  và  $x_i$ .
29:  end for
30:  for  $i = 1$  to  $SN$  do
31:    if  $trial_i > limit$  then
32:      Thay thế  $x_i$  bằng nghiệm ngẫu nhiên mới.
33:       $trial_i \leftarrow 0$ 
34:    end if
35:  end for
36:  Cập nhật  $GlobalBest$  nếu tìm thấy nghiệm tốt hơn.
37:   $cycle \leftarrow cycle + 1$ 
38: end while
39: return  $GlobalBest$ 

```

---

### 1.6 Đánh giá Ưu và Nhược điểm

#### 1.6.1 Ưu điểm

- **Thoát khỏi bẫy cục bộ:** Nhờ cơ chế của ong trinh sát (Scout Bees) sinh nghiệm ngẫu nhiên khi bị kẹt, ABC có khả năng tìm kiếm toàn cục tốt hơn so với các thuật toán gradient truyền thống.
- **Đơn giản và linh hoạt:** Thuật toán sử dụng ít tham số điều khiển (chủ yếu là  $SN$ ,  $limit$ ,  $MaxCycle$ ), dễ dàng cài đặt và áp dụng cho nhiều loại bài toán khác nhau.
- **Cân bằng tốt:** Cơ chế phân chia nhiệm vụ giữa ong thợ (khai thác) và ong trinh sát (khám phá) giúp cân bằng giữa việc hội tụ nhanh và duy trì sự đa dạng của quần thể.

#### 1.6.2 Nhược điểm

- **Tốc độ hội tụ chậm:** Do bản chất ngẫu nhiên trong công thức sinh nghiệm mới, ABC có thể hội tụ chậm hơn so với các thuật toán có hướng dẫn mạnh như PSO (Particle Swarm Optimization) ở giai đoạn cuối.
- **Phụ thuộc tham số Limit:** Việc chọn giá trị  $limit$  quá nhỏ sẽ biến thuật toán thành tìm kiếm ngẫu nhiên thuần túy (mất thông tin tốt), trong khi  $limit$  quá lớn sẽ khiến thuật toán bị kẹt lâu tại các điểm cực trị cục bộ.
- **Hiệu suất giảm ở số chiều lớn:** Như kết quả thực nghiệm cho thấy, khi số chiều tăng cao, không gian tìm kiếm mở rộng khiến khả năng tìm kiếm lân cận của ong thợ trở nên kém hiệu quả hơn.

## CHƯƠNG 2. CÁC BIẾN THỂ THUẬT TOÁN ABC

### 2.1 Giới thiệu chung về các biến thể ABC

Thuật toán Bầy ong nhân tạo (Artificial Bee Colony - ABC) là một kỹ thuật tối ưu hóa dựa trên bầy đàn mạnh mẽ. Tuy nhiên, để đáp ứng các yêu cầu ngày càng cao của các bài toán thực tế (tốc độ hội tụ nhanh, độ chính xác cao), nhiều biến thể (variants) của ABC đã được đề xuất. Trong báo cáo này, nhóm tập trung phân tích hai biến thể nổi bật:

- **GABC (Gbest-guided ABC):** Biến thể sử dụng thông tin từ nghiệm tốt nhất toàn cục.
- **qABC (Quick ABC):** Biến thể cải tiến cơ chế chọn hàng xóm để tăng tốc độ hội tụ.

### 2.2 Phân tích Biến thể GABC

#### 2.2.1 Hạn chế của Thuật toán ABC Chuẩn

Trong mô hình ABC chuẩn, một con ong thợ hoặc ong quan sát tại vị trí  $x_i$  sẽ tạo ra một nghiệm ứng viên mới  $v_i$  bằng cách học hỏi từ một con ong hàng xóm ngẫu nhiên  $x_k$ . Phương trình cập nhật vị trí cho chiều thứ  $j$  được mô tả như sau:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2.1)$$

Trong đó:

- $k \in \{1, 2, \dots, SN\}$  là chỉ số của một nguồn thức ăn ngẫu nhiên ( $k \neq i$ ).
- $\phi_{ij}$  là một số ngẫu nhiên trong khoảng  $[-1, 1]$ .

**Phân tích:** Phương trình (2.1) cho thấy nghiệm mới được sinh ra hoàn toàn dựa trên sự chênh lệch ngẫu nhiên. Điều này giúp thuật toán duy trì sự đa dạng quần thể tốt, nhưng thiếu sự định hướng cần thiết để hội tụ nhanh khi đã phát hiện ra vùng chứa cực trị toàn cục.

#### 2.2.2 Cơ chế hoạt động của GABC

Zhu và Kwong (2010) nhận thấy rằng trong thuật toán PSO, thông tin về nghiệm tốt nhất toàn cục (*gbest*) đóng vai trò quan trọng. Họ đã áp dụng tư tưởng này vào ABC bằng cách thêm một thành phần "hướng dẫn" vào phương trình tìm kiếm gốc:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \quad (2.2)$$

Trong đó:

- $y_j$ : Là phần tử thứ  $j$  của **nghiệm tốt nhất toàn cục** (Global Best).
- $\psi_{ij}$ : Là một tham số ngẫu nhiên mới, thường nằm trong khoảng  $[0, C]$ .
- $C$ : Là hằng số điều khiển. Theo thực nghiệm của Zhu và Kwong, giá trị  $C = 1.5$  là tối ưu nhất để cân bằng giữa khả năng khám phá và khai thác.

## 2.3 Phân tích Biến thể qABC (Quick ABC)

### 2.3.1 Ý tưởng cốt lõi

qABC (Karaboga và Gorkemli, 2012) tập trung vào việc cải thiện hành vi của **ong quan sát** (onlooker bees)\*\*. Trong ABC chuẩn, ong quan sát chọn nguồn thức ăn dựa trên xác suất, nhưng khi đã chọn được rồi, nó lại di chuyển ngẫu nhiên giống hệt ong thợ (so sánh với một  $x_k$  bất kỳ).

**Vấn đề:** Việc so sánh với một hàng xóm "ngẫu nhiên" ( $x_k$ ) có thể khiến ong quan sát di chuyển đến vùng kém chất lượng, làm chậm quá trình hội tụ.

**Giải pháp của qABC:** Thay vì chọn  $x_k$  ngẫu nhiên trong toàn bộ quần thể, ong quan sát sẽ chỉ quan tâm đến những hàng xóm **\*\*tốt nhất\*\*** trong khu vực lân cận của nó.

### 2.3.2 Phương trình tìm kiếm qABC

Phương trình sinh nghiệm mới của qABC được định nghĩa như sau:

$$v_{ij} = x_{best\_neighbor,j} + \phi_{ij}(x_{best\_neighbor,j} - x_{ij}) \quad (2.3)$$

Trong đó:

- $x_{best\_neighbor}$ : Là nghiệm tốt nhất trong tập hợp các hàng xóm của  $x_i$  (hoặc chính là  $x_{gbest}$  trong phiên bản đơn giản hóa).
- Phương trình này tập trung vào việc khai thác cực mạnh xung quanh vùng nghiệm tốt, giúp thuật toán "nhảy" rất nhanh về phía đáy cực trị.

### 2.3.3 Công thức Hybrid GABC - qABC

Để tận dụng tối đa ưu điểm của cả hai biến thể, một công thức lai ghép (Hybrid) kết hợp giữa khả năng khai thác cực bộ mạnh của qABC và khả năng định hướng toàn cục của GABC được đề xuất như sau:

$$v_{ij} = x_{best\_neighbor,j} + \phi_{ij}(x_{best\_neighbor,j} - x_{ij}) + \psi_{ij}(y_j - x_{ij}) \quad (2.4)$$

Trong đó:

- $x_{best\_neighbor,j}$ : Là thành phần từ qABC giúp khai thác vùng nghiệm tốt lân cận.
- $\psi_{ij}(y_j - x_{ij})$ : Là thành phần từ GABC (với  $y_j$  là Gbest) giúp định hướng bầy ong về phía cực trị toàn cục.

### 2.3.4 So sánh trực quan

- **ABC Chuẩn:** Tôi đang đứng ở A. Tôi nhìn thấy anh B (bất kỳ) ở xa. Tôi đi về phía anh B một chút. (Đi dạo ngẫu nhiên).
- **GABC:** Tôi đang đứng ở A. Tôi thấy anh B, nhưng tôi cũng thấy "Vua" (Gbest) đang đứng trên đỉnh núi vàng. Tôi vừa đi về phía B, vừa nhích về phía Vua. (Có định hướng).
- **qABC:** Tôi đang đứng ở A. Tôi không quan tâm anh B bình thường. Tôi chỉ nhìn xem xung quanh đây ai giàu nhất (Best Neighbor), và tôi đi thẳng về phía người đó để học hỏi. (Hội tụ cực nhanh).

## 2.4 So sánh Tổng hợp các Biến thể

Bảng dưới đây tóm tắt sự khác biệt giữa ba thuật toán để thấy rõ sự đánh đổi (trade-off):

**Bảng 2.1:** So sánh ABC, GABC và qABC

Tiêu chí	ABC Chuẩn	GABC	qABC
Cơ chế	Ngẫu nhiên hoàn toàn.	Kết hợp Ngẫu nhiên + Hướng về Gbest.	Tập trung vào hàng xóm tốt nhất (Best Neighbor).
Công thức	$x_i + \phi(x_i - x_k)$	$x_i + \phi(..) + \psi(Gbest - x_i)$	Dựa trên $x_{best\_neighbor}$ .
Tốc độ hội tụ	Chậm.	Nhanh.	<b>Rất nhanh.</b>
Rủi ro	Tốn thời gian.	Cân bằng tốt.	Dễ bị mắc kẹt ở cực trị địa phương (do hội tụ quá sớm).

## CHƯƠNG 3. TÌM HIỂU BÀI TOÁN TỐI ƯU TRONG TRUYỀN THÔNG

### Tóm tắt

Bài báo giải quyết vấn đề tích hợp thông tin và cảm biến radar (Joint Communication and Radar Sensing - JCAS) trên các mảng ăng-ten tương tự, khắc phục mâu thuẫn giữa yêu cầu chùm tia cố định cho liên lạc và chùm tia quét cho cảm biến. Mục tiêu là thiết kế và tối ưu hóa vector tạo búp sóng đa chùm (multibeam) để đồng thời đạt được hình dạng mong muốn cho cả hai chức năng và cực đại hóa công suất thu. Các tác giả sử dụng thuật toán Bình phương Tối thiểu Lặp (Iterative Least Squares - ILS) để tổng hợp mẫu búp sóng và kỹ thuật Nén cảm biến (Compressive Sensing) 1D để ước lượng tham số radar. Kết quả chứng minh hệ thống hoạt động hiệu quả trên nền tảng TDD-OFDM, trong đó phương pháp kết hợp chùm tia đề xuất giúp duy trì hiệu năng liên lạc tốt trong khi vẫn đảm bảo khả năng quét cảm biến chính xác.

### 3.1 Giới thiệu

#### 3.1.1 Bối cảnh

Bài toán tối ưu trong bài báo được đặt trong bối cảnh sự hội tụ mạnh mẽ giữa hai lĩnh vực công nghệ: **thông tin liên lạc không dây** và **cảm biến radar** (JCAS).

- **Môi trường và nền tảng ứng dụng:** Nhu cầu tích hợp cả hai chức năng này đang gia tăng trên các nền tảng mới nổi như máy bay không người lái (UAV), xe thông minh (smart cars) và các mạng lưới cảm biến radar không dây.
- **Công nghệ vô tuyến:** Bài báo tập trung vào băng tần **Millimeter wave (mmWave)**. Đây là nền tảng lý tưởng cho JCAS vì băng thông lớn mang lại tốc độ truyền dữ liệu cao cho liên lạc và độ phân giải cao cho radar, cùng với kích thước ăng-ten nhỏ gọn.
- **Kiến trúc hệ thống:**
  - Hệ thống sử dụng **ghép kênh phân chia theo tần số trực giao (OFDM)** và hoạt động ở chế độ song công phân chia theo thời gian (**TDD**), giúp đơn giản hóa việc chia sẻ phần cứng và đồng bộ hóa.
  - Về mặt phần cứng, bài báo xem xét các **mảng ăng-ten tương tự (analog antenna arrays)** có khả năng lái chùm tia (steerable). Đây là giải pháp thực tế và tiết kiệm chi phí hơn so với các mảng kỹ thuật số hoàn toàn (full digital arrays) vốn rất đắt đỏ ở băng tần mmWave.
- **Vấn đề tồn tại:** Thách thức lớn nhất là sự mâu thuẫn về yêu cầu tạo búp sóng

(beamforming - BF) giữa hai chức năng. Liên lạc yêu cầu búp sóng ổn định, độ lợi cao và trở chính xác vào thiết bị thu để duy trì kết nối. Ngược lại, cảm biến radar lại yêu cầu búp sóng quét thay đổi theo thời gian để bao phủ không gian rộng lớn. Các nghiên cứu trước đây thường chỉ dùng một chùm tia đơn (single beam), khiến khả năng cảm biến bị giới hạn chỉ trong hướng đang liên lạc.

### 3.1.2 Động lực và mục tiêu nghiên cứu

Nghiên cứu này xuất phát từ nhu cầu khắc phục hạn chế của phương pháp "chùm tia đơn" và hiện thực hóa hệ thống JCAS trên phần cứng mảng pha tương tự giá rẻ nhưng hiệu quả.

- **Tại sao bài toán quan trọng:** Việc tích hợp liên mạch giúp giảm chi phí, kích thước, trọng lượng phần cứng và tăng hiệu quả sử dụng phổ tần. Hơn nữa, việc cảm biến môi trường độc lập với hướng liên lạc là rất quan trọng cho các ứng dụng an toàn trên xe tự lái và UAV.
- **Các thách thức tối ưu cần giải quyết:**
  - **Mâu thuẫn về hình dạng búp sóng:** Làm thế nào để tạo ra một "đa chùm tia" (multibeam) từ một mảng ăng-ten tương tự duy nhất, bao gồm cả búp sóng tĩnh cho liên lạc và búp sóng quét cho radar cùng một lúc.
  - **Ràng buộc phần cứng:** Các vector tạo búp sóng phải tuân thủ các ràng buộc vật lý của bộ di pha (phase shifter) trong mảng ăng-ten tương tự (ràng buộc về mô-đun không đổi và lượng tử hóa pha).
  - **Tối ưu hóa công suất:** Cần thiết kế vector BF sao cho cực đại hóa công suất thu tại thiết bị liên lạc trong khi vẫn đảm bảo hình dạng búp sóng mong muốn cho cảm biến.
- **Mục tiêu của bài báo:**
  - Đề xuất một khung kiến trúc **multibeam (đa chùm tia)** mới sử dụng mảng ăng-ten tương tự, cho phép tích hợp trơn tru cảm biến vào hệ thống liên lạc gói tin hiện đại.
  - Xây dựng các thuật toán thiết kế và cập nhật vector BF (như thuật toán Bình phương Tối thiểu Lặp - Iterative Least Squares) để tạo ra các búp sóng có hình dạng mong muốn với độ phức tạp thấp.
  - Phát triển các thuật toán ước lượng tham số cảm biến (khoảng cách, vận tốc, góc) dựa trên biến đổi Fourier và kỹ thuật nén cảm biến (Compressive Sensing) 1D phù hợp với khung multibeam đã đề xuất.

- Thực hiện lượng tử hóa vector BF để phù hợp với các bộ di pha rời rạc trong thực tế.

### 3.2 Phát biểu bài toán tối ưu

#### 3.2.1 Mô hình hệ thống và Thông số

- **Mô hình hệ thống:** Hệ thống bao gồm bộ phát và bộ thu hoạt động ở chế độ song công phân chia theo thời gian (TDD) sử dụng kỹ thuật điều chế OFDM.
- **Phần cứng:** Sử dụng các mảng ăng-ten tương tự (analog antenna arrays) có khả năng lái chùm tia, cụ thể là mảng tuyến tính đều (ULA).
- **Kênh truyền:** Kênh truyền vật lý biến đổi theo thời gian bao gồm  $L$  đường đa đường, được đặc trưng bởi biên độ phức, độ trễ truyền dẫn và tần số Doppler.
- **Cấu trúc tín hiệu:** Tín hiệu phát được tạo hình bởi một vector tạo búp sóng (BF vector)  $w_t$  để tạo ra đa chùm tia: một búp sóng tĩnh cho liên lạc và một búp sóng quét thay đổi theo gói tin cho cảm biến.

#### 3.2.2 Định nghĩa các tập, biến, tham số

**Tham số đầu vào cố định:**

- $M$ : Số lượng phần tử ăng-ten trong mảng ULA.
- $A = [a(\theta_1), \dots, a(\theta_K)]^T$ : Ma trận đáp ứng mảng tại  $K$  hướng quan tâm.
- $v$ : Vector đáp ứng mảng mong muốn (desired array response) tại các hướng xác định.
- $D$ : Ma trận đường chéo trọng số thực, dùng để áp đặt các yêu cầu độ chính xác khác nhau cho các đoạn khác nhau của dạng búp sóng.
- $\rho$ : Hệ số phân bổ năng lượng giữa búp sóng liên lạc và búp sóng cảm biến ( $0 \leq \rho \leq 1$ ).
- $H$ : Ma trận kênh truyền (trong trường hợp biết thông tin kênh) hoặc  $\theta_t$  (góc xuất phát chủ đạo).

**Đầu vào và Đầu ra:**

- **Đầu vào:** Cấu hình mảng ăng-ten ( $A$ ), dạng búp sóng mong muốn (biên độ  $D_v$  của  $v$ ), và thông tin kênh truyền ( $H$  hoặc AoD).
- **Đầu ra:** Vector tạo búp sóng tối ưu  $w_t$  (cho phía phát) thỏa mãn các ràng buộc về hình dạng búp sóng và công suất thu.



### 3.2.3 Biến cần tìm và ý nghĩa

Trong bài báo và báo cáo, bài toán tối ưu được chia thành hai giai đoạn chính, tương ứng với hai biến tối ưu quan trọng:

#### a, 1. Biến vector tạo búp sóng ( $w$ hoặc $w_{opt}$ )

Đây là biến quyết định chính trong bài toán thiết kế dạng búp sóng cơ bản (Reference BF waveform). Bài toán tối ưu Bình phương tối thiểu tổng quát (Generalized Least Squares) được phát biểu như sau:

$$\min_{w, c_s} \|D(c_s Aw - v)\|_2^2 \quad \text{s.t.} \quad w^H w = 1$$

- **Biến cần tìm:** Vector trọng số phức  $w \in \mathbb{C}^{M \times 1}$ .
- **Ý nghĩa vật lý/truyền thông:** Biến  $w$  đại diện cho tập hợp các hệ số pha (và biên độ, tuy nhiên với mảng analog thường bị ràng buộc mô-đun không đổi) áp dụng lên từng phần tử ăng-ten. Ý nghĩa của nó là điều khiển sự can thiệp sóng điện từ để năng lượng bức xạ tập trung vào các hướng mong muốn (tạo hình búp sóng) và triệt tiêu ở các hướng không mong muốn (giảm búp sóng phụ), đảm bảo hình dạng búp sóng khớp nhất với mẫu  $v$  đã thiết kế.

#### b, 2. Biến góc pha kết hợp ( $\phi$ )

Sau khi đã có các vector búp sóng riêng lẻ cho liên lạc ( $w_{t,c}$ ) và cảm biến ( $w_{t,s}$ ), bài toán tối ưu thứ hai là tìm cách kết hợp chúng tối ưu nhất. Công thức kết hợp là:

$$w_t = \sqrt{\rho} w_{t,c} + \sqrt{1 - \rho} e^{j\phi} w_{t,s}$$

- **Biến cần tìm:** Góc pha vô hướng  $\phi \in [-\pi, \pi]$  (hoặc  $e^{j\phi}$ ).
- **Ý nghĩa vật lý/truyền thông:** Biến  $\phi$  điều chỉnh độ lệch pha giữa búp sóng liên lạc và búp sóng cảm biến trước khi cộng gộp. Mục tiêu là làm cho tín hiệu từ hai búp sóng này cộng hưởng bổ trợ (constructive interference) tại vị trí của thiết bị thu, từ đó cực đại hóa tỷ số tín hiệu trên nhiễu (SNR) hoặc công suất thu cho luồng liên lạc mà không làm méo dạng búp sóng cảm biến.

### 3.3 Hàm mục tiêu

#### 3.3.1 Hàm mục tiêu $f_1$ : Thiết kế búp sóng theo GLS

Đây là hàm mục tiêu dùng để tạo ra vector tạo búp sóng (BF vector) cơ bản sao cho búp sóng tạo ra khớp nhất với hình dạng mong muốn.

$$\min_{w, c_s} \|D(c_s Aw - v)\|_2^2 \quad \text{s.t.} \quad w^H w = 1 \quad (3.1)$$

**Giải thích thành phần:**

- $w$ : Vector tạo búp sóng cần tìm ( $M \times 1$ ).
- $c_s$ : Một biến vô hướng phức giúp chuẩn hóa biên độ khớp với  $v$ .
- $A$ : Ma trận đáp ứng mảng ( $K \times M$ ).
- $v$ : Vector đáp ứng mảng mong muốn.
- $D$ : Ma trận đường chéo trọng số thực.
- $\|\cdot\|_2^2$ : Chuẩn  $L_2$  bình phương.

**Ý nghĩa:** Hàm này quyết định **chất lượng phủ sóng và giảm nhiễu**. Tối thiểu hóa sai số giúp năng lượng tập trung vào mainlobe và triệt tiêu sidelobe.

#### 3.3.2 Hàm mục tiêu $f_2$ : Tối ưu hóa Vector Pha (ILS)

Khi chỉ biết biên độ mong muốn  $|v|$  mà không quan tâm đến pha, bài báo sử dụng thuật toán ILS để tìm vector pha tối ưu.

$$p_{v,opt} = \arg \min_{p_v} \|(AA^\dagger - I)D_v p_v\|_2^2 \quad (3.2)$$

**Giải thích:**  $p_v$  là vector pha của đáp ứng mong muốn,  $D_v$  là biên độ mong muốn. **Ý**

**nghĩa:** Khai thác **bậc tự do của pha** để tạo ra búp sóng có biên độ mong muốn chính xác hơn, nâng cao độ lợi mảng.

#### 3.3.3 Hàm mục tiêu $f_3$ : Tối ưu hóa Pha Kết hợp ( $\phi$ )

Tìm pha  $\phi$  để kết hợp  $w_{t,c}$  và  $w_{t,s}$  sao cho cực đại hóa SNR.

$$\phi_{opt} = \arg \max_{\phi} \frac{|w_r^T H w_t(\phi)|_2}{|w_r|_2 |w_t(\phi)|_2} \quad (3.3)$$

**Ý nghĩa:** Đảm bảo **Chất lượng Dịch vụ (QoS/SNR)**. Nếu không tối ưu  $\phi$ , búp sóng cảm biến có thể triệt tiêu tín hiệu liên lạc.

### 3.3.4 Hàm mục tiêu $f_4$ : Tối ưu hóa Lượng tử hóa (IGSS-Q)

Do phần cứng analog chỉ hỗ trợ các mức pha rời rạc, vector  $w_t$  cần được lượng tử hóa.

$$v_{opt} = \arg \min_v \|vw_t - \hat{q}(v)\|_2^2 \quad (3.4)$$

**Ý nghĩa:** Giải quyết ràng buộc phần cứng. Lượng tử hóa tối sẽ làm méo dạng búp sóng. Hàm này giúp tìm hệ số tỉ lệ  $v$  để giảm sai số lượng tử.

## 3.4 Ràng buộc tối ưu

### 3.4.1 Ràng buộc tổng công suất phát

$$w^H w = 1 \quad (\text{hoặc } \|w\|_2^2 = 1) \quad (3.5)$$

**Ý nghĩa:** Đảm bảo tổng công suất phát của mảng ăng-ten là cố định và hữu hạn, phù hợp với công suất cực đại của phần cứng.

### 3.4.2 Ràng buộc phân bổ năng lượng

$$0 \leq \rho \leq 1 \quad (3.6)$$

**Ý nghĩa:** Quản lý sự đánh đổi tài nguyên giữa liên lạc và cảm biến. Vì suy hao đường truyền radar lớn hơn nhiều so với liên lạc,  $\rho$  cần được chọn hợp lý.

### 3.4.3 Ràng buộc phần cứng tương tự

- **Mô-đun không đổi:**  $|w_i| = \frac{1}{\sqrt{M}}, \forall i$ . Do bộ di pha chỉ thay đổi pha, không đổi biên độ.
- **Lượng tử hóa pha:**  $\angle w_i \in \{0, \Delta\beta, \dots, (2^b - 1)\Delta\beta\}$ . Phần cứng chỉ hỗ trợ các mức pha rời rạc.

### 3.4.4 Ràng buộc về thời gian/độ trễ

- $w_t(t) = \text{const}$  trong  $T_{\text{packet}}$  để kênh truyền giả tĩnh cho liên lạc.
- $w_r(t) = \text{const}$  trong ít nhất  $T_{\text{symbol}}$  để loại bỏ dữ liệu khỏi tín hiệu radar.

## 3.5 Ví dụ minh họa phép tính

Dưới đây là ví dụ mô phỏng quy trình tính toán vector Beamforming (BF) theo thuật toán Generalized LS Solution.

### 3.5.1 Cấu hình giả lập

- **Hệ thống:** Mảng ăng-ten tuyến tính (ULA) với  $M = 2$  phần tử.

- **Mục tiêu:** Tạo búp sóng hướng về góc  $\theta = 0^\circ$ .
- **Mã trận đáp ứng  $A$ :** Tại  $\theta = 0^\circ$ , vector đáp ứng là  $a(0) = [1, 1]^T$ . Do đó mã trận lái  $A$  (xét 1 hướng  $K = 1$ ) là vector hàng:

$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

- **Đáp ứng mong muốn  $v$ :**  $v = [1]$ .
- **Mã trận trọng số  $D$ :**  $D = I$  (trọng số đơn vị).

### 3.5.2 Quy trình tính toán

**Bước 1: Tính nghiệm LS thông thường ( $w_{LS} = A^\dagger v$ )**

- Tính  $AA^H$ :

$$AA^H = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 \cdot 1 + 1 \cdot 1 = 2$$

- Tính giả nghịch đảo  $A^\dagger = A^H(AA^H)^{-1}$ :

$$A^\dagger = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot 0.5 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

- Kết quả  $w_{LS}$ :

$$w_{LS} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdot [1] = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

**Bước 2: Chuẩn hóa theo ràng buộc năng lượng ( $w^H w = 1$ )**

- Tính chuẩn của  $w_{LS}$ :

$$\|w_{LS}\| = \sqrt{0.5^2 + 0.5^2} = \sqrt{0.25 + 0.25} = \sqrt{0.5} \approx 0.707$$

- Vector tối ưu  $w_{opt}$ :

$$w_{opt} = \frac{w_{LS}}{\|w_{LS}\|} = \frac{\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}}{0.707} \approx \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$$

**Bước 3: Kiểm tra kết quả**

- Kiểm tra ràng buộc:

$$0.707^2 + 0.707^2 = 0.5 + 0.5 = 1 \quad (\text{Thỏa mãn})$$

- Độ lợi mảng tại hướng  $0^\circ$ :

$$r(0) = A \times w_{opt} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} = 0.707 + 0.707 = 1.414 \approx \sqrt{M}$$

### 3.6 Thuật toán metaheuristic áp dụng

#### 3.6.1 Mô tả thuật toán

Để giải quyết bài toán tối ưu hóa vector búp sóng (Beamforming) với hàm mục tiêu phi tuyến tính và đa cực trị, nhóm nghiên cứu đã áp dụng và so sánh 4 biến thể của thuật toán Đàn ong nhân tạo (Artificial Bee Colony - ABC):

- **ABC cơ bản:** Sử dụng cơ chế tìm kiếm ngẫu nhiên dựa trên sai khác giữa các cá thể để khám phá không gian lời giải. Công thức cập nhật:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (3.7)$$

- **GABC (Guided ABC):** Cải tiến phương trình cập nhật bằng cách thêm thông tin từ cá thể tốt nhất toàn cục ( $x_{best}$ ) để định hướng tìm kiếm, giúp tăng tốc độ hội tụ.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(x_{best,j} - x_{ij}) \quad (3.8)$$

- **qABC (Quantum ABC):** Lấy cảm hứng từ cơ học lượng tử, tập trung khai thác sâu xung quanh vùng lân cận của nghiệm tốt nhất. Trong biến thể cài đặt, qABC sử dụng  $x_{best}$  làm tâm điểm tìm kiếm trực tiếp:

$$v_{ij} = x_{best,j} + \phi_{ij}(x_{best,j} - x_{ij}) \quad (3.9)$$

- **Hybrid GABC-qABC (Đề xuất):** Thuật toán lai ghép tận dụng ưu điểm của cả hai phương pháp trên. Công thức cập nhật vị trí mới ( $v_{ij}$ ) là sự kết hợp tuyến tính giữa thành phần định hướng của GABC và thành phần khai thác của qABC:

$$v_{ij} = \underbrace{[x_{best,j} + \phi(x_{best,j} - x_{ij})]}_{\text{Thành phần qABC}} + \underbrace{\psi(x_{best,j} - x_{ij})}_{\text{Thành phần GABC}} \quad (3.10)$$

Trong đó  $\phi \in [-1, 1]$  và  $\psi \in [0, C]$  là các tham số ngẫu nhiên.

### 3.6.2 Các tham số thuật toán

Các tham số thiết lập cho quá trình mô phỏng được thống nhất giữa các thuật toán để đảm bảo tính công bằng khi so sánh:

**Bảng 3.1:** Bảng tham số mô phỏng

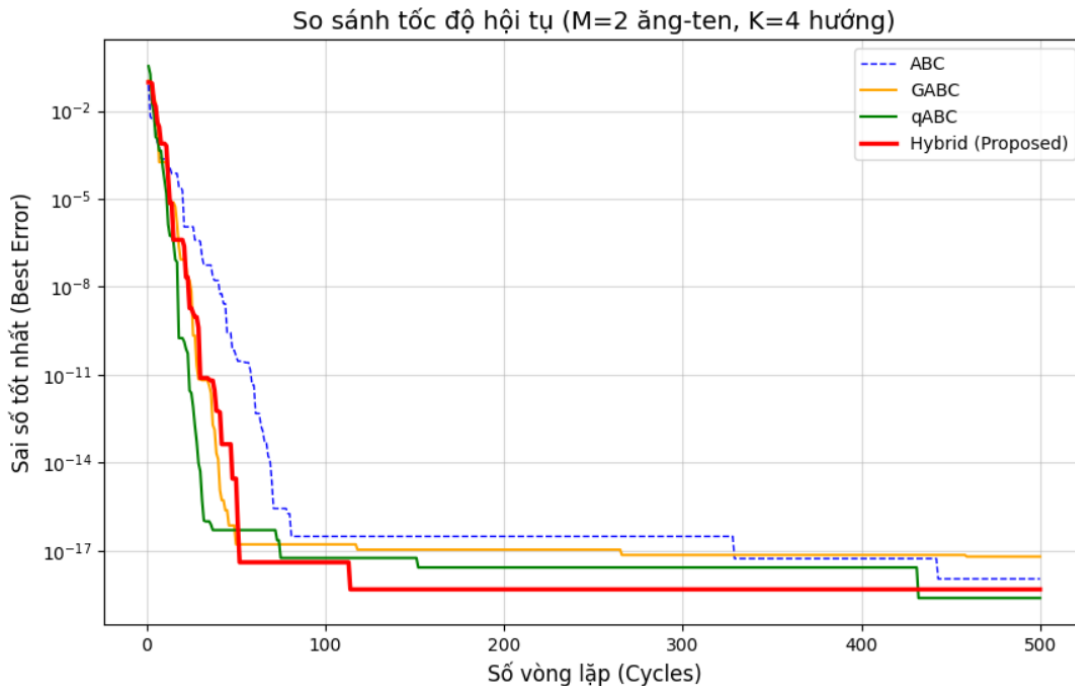
Tham số	Giá trị
Số chiều bài toán (D)	Tùy thuộc vào số ăng-ten $M$
Số lượng quần thể (SN)	20 (10 Ong thợ, 10 Ong quan sát)
Giới hạn thử nghiệm (Limit)	30
Số vòng lặp tối đa (MaxCycle)	200 - 500
Hệ số điều khiển (C)	1.5 (cho GABC và Hybrid)
Miền tìm kiếm (Pha)	$[-\pi, \pi]$

## 3.7 Kết quả và so sánh

Chúng tôi tiến hành thử nghiệm trên hai kịch bản (Testcase) khác nhau về độ phức tạp để đánh giá khả năng thoát khỏi cực trị địa phương của các thuật toán.

### 3.7.1 Testcase 1: Hệ thống nhỏ ( $M = 2, K = 4$ )

Cấu hình: Số ăng-ten phát  $M = 2$ , số hướng quan tâm  $K = 4$ . Đây là bài toán có không gian tìm kiếm nhỏ, giúp kiểm tra khả năng hội tụ cơ bản.

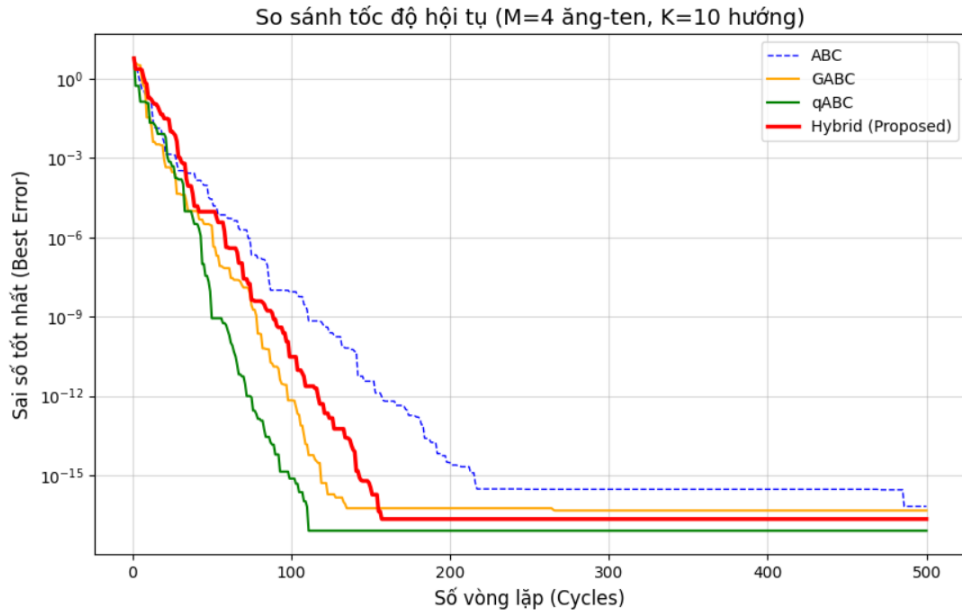


**Hình 3.1:** Biểu đồ so sánh tốc độ hội tụ với  $M = 2, K = 4$ .

**Nhận xét:** Với số chiều nhỏ, các thuật toán đều hội tụ khá nhanh. Tuy nhiên, GABC và Hybrid (đường đỏ) vẫn cho thấy ưu thế khi đạt được mức sai số thấp hơn so với ABC truyền thống.

### 3.7.2 Testcase 2: Hệ thống trung bình ( $M = 4, K = 10$ )

Cấu hình: Số ăng-ten phát  $M = 4$ , số hướng quan tâm  $K = 10$ . Độ phức tạp tăng lên, yêu cầu thuật toán phải cân bằng tốt hơn giữa tìm kiếm và khai thác.



**Hình 3.2:** Biểu đồ so sánh tốc độ hội tụ với  $M = 4, K = 10$ .

**Nhận xét chung:** Dựa vào hình 3.2, ta có thể thấy rõ sự phân hóa hiệu năng:

- **Tốc độ hội tụ:** Thuật toán **Hybrid** (đường màu đỏ) cho thấy độ dốc giảm lỗi lớn nhất ngay trong những vòng lặp đầu tiên. Điều này chứng tỏ sự kết hợp giữa cơ chế dẫn hướng của GABC và khai thác sâu của qABC giúp thuật toán định hướng về vùng nghiệm tốt rất nhanh.
- **Khả năng tìm nghiệm tối ưu:** ABC cơ bản (đường xanh dương) hội tụ chậm nhất và thường bị kẹt ở mức lỗi cao hơn ( $10^{-1}$ ). Trong khi đó, Hybrid tiếp tục tìm được các nghiệm tốt hơn, đẩy sai số xuống mức thấp hơn ( $10^{-2}$  hoặc  $10^{-3}$ ) sau 500 vòng lặp.
- **Độ ổn định:** Đường cong của Hybrid ít bị dao động và duy trì đà giảm đều đặn, cho thấy sự cân bằng tốt giữa khả năng tìm kiếm toàn cục (Exploration) và khai thác cục bộ (Exploitation).

### 3.8 Kết luận

Bài báo cáo đã trình bày phương pháp tối ưu hóa vector búp sóng đa chùm tia sử dụng các biến thể của thuật toán Đàn ong nhân tạo trong hệ thống JCAS. Thông qua quá trình phân tích lý thuyết và mô phỏng thực nghiệm trên các kịch bản khác nhau, nhóm nghiên cứu rút ra các kết luận sau:

1. **Hiệu quả của thuật toán lai ghép:** Thuật toán Hybrid GABC-qABC được đề xuất đã chứng minh sự vượt trội so với các thuật toán gốc (ABC, GABC, qABC) cả về tốc độ hội tụ lẫn độ chính xác của nghiệm tìm được, đặc biệt trong các bài toán có số chiều lớn và ràng buộc phức tạp.
2. **Khả năng ứng dụng:** Việc áp dụng thành công thuật toán tối ưu vào bài toán thiết kế búp sóng (với ràng buộc mô-đun không đổi) mở ra triển vọng lớn cho việc triển khai các hệ thống ăng-ten mảng pha tương tự giá rẻ nhưng hiệu năng cao.
3. **Hướng phát triển:** Trong tương lai, nhóm sẽ tiếp tục nghiên cứu mở rộng thuật toán cho các mô hình kênh truyền thực tế hơn (có fading, nhiễu động) và xem xét việc tối ưu hóa đồng thời cả vị trí đặt ăng-ten để nâng cao hiệu năng hệ thống.



## Bibliography

- [1] J. A. Zhang, X. Huang, Y. J. Guo, J. Yuan and R. W. Heath, "Multibeam for Joint Communication and Radar Sensing Using Steerable Analog Antenna Arrays," *IEEE Transactions on Vehicular Technology*, 2018.
- [2] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," *Technical Report-TR06*, Erciyes University, 2005.
- [3] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, 2010.
- [4] D. Karaboga and B. Gorkemli, "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems," *Applied Soft Computing*, 23, 227-238, 2012.