

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



BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: NHẬP MÔN KỸ THUẬT TRUYỀN THÔNG

THUẬT TOÁN BẦY ONG NHÂN
TẠO (ABC) VÀ CÁC BIẾN THỂ

STT	Họ và tên	Mã số sinh viên
1	Nguyễn Đức Hiếu	20235712
2	Nguyễn Minh Xuân	20235882

Trợ giảng hướng dẫn: Nguyễn Quang Đông
Vũ Minh Quân

Hà Nội, ngày 30 tháng 11 năm 2025

Mục lục

1	Giới thiệu chung	2
1.1	Bối cảnh	2
1.2	Mục tiêu nghiên cứu	2
1.3	Cấu trúc báo cáo	2
2	Thuật toán Artificial Bee Colony (ABC)	3
2.1	Nguồn gốc và lịch sử	3
2.2	Phân loại và đặc tính thuật toán	3
2.2.1	Phân loại	3
2.2.2	Ưu điểm và Nhược điểm	3
2.3	Nguyên lý hoạt động của đàn ong mật (Cơ chế tự nhiên)	3
2.3.1	Ba loại ong chức năng	4
2.3.2	Cơ chế chia sẻ thông tin (Waggle Dance)	4
2.3.3	Ảnh xạ từ Tự nhiên sang Thuật toán	4
2.4	Mô hình toán học của thuật toán ABC	4
2.4.1	Công thức tìm kiếm lân cận (Sinh nghiệm mới)	4
2.4.2	Hàm đánh giá Fitness	5
2.4.3	Công thức tính xác suất chọn lựa	5
2.5	Mã giả và quy trình hoạt động	5
3	Các biến thể của thuật toán ABC	6
3.1	Gbest-guided Artificial Bee Colony (GABC)	6
3.1.1	Ý tưởng chủ đạo	6
3.1.2	Phương trình cải tiến	6
3.1.3	Phân tích sự khác biệt về cơ chế hoạt động	6
3.1.4	Vai trò quan trọng của tham số điều chỉnh C	7
3.2	Quick Artificial Bee Colony (qABC)	7
3.2.1	Nguồn gốc và Cảm hứng sinh học	7
3.2.2	Cơ chế “Hàng xóm tinh anh” (Elite Neighbor Strategy)	7
3.2.3	Ảnh hưởng của Bán kính lân cận (r)	8
4	Mô phỏng và đánh giá hiệu năng	9
4.1	Thí nghiệm 1: Đánh giá tốc độ hội tụ trên hàm Sphere	9
4.1.1	Thiết lập bài toán	9
4.1.2	Kết quả và Thảo luận	9
4.2	Thí nghiệm 2: Đánh giá tốc độ hội tụ trên hàm Shifted Rastrigin	11
4.2.1	Thiết lập bài toán	11
4.2.2	Kết quả và Thảo luận	13
5	ỨNG DỤNG TRONG KỸ THUẬT TRUYỀN THÔNG	14
5.1	Tối ưu hóa tài nguyên mạng (Resource Allocation)	14
5.2	Ước lượng kênh truyền trong hệ thống MIMO	14
5.3	Định tuyến trong Mạng cảm biến không dây (WSN)	14
6	KẾT LUẬN	14

Tóm tắt nội dung

Báo cáo này trình bày nghiên cứu chi tiết về thuật toán Bầy ong nhân tạo (Artificial Bee Colony – ABC), một phương pháp tối ưu hóa metaheuristic dựa trên trí tuệ bầy đàn, mô phỏng hành vi tìm kiếm thức ăn thông minh của loài ong mật. Báo cáo phân tích toàn diện cơ chế hoạt động của ABC, bao gồm mô hình toán học và sự phối hợp giữa ba loại tác nhân chính: ong thợ, ong quan sát và ong trình sát, nhằm giải quyết bài toán cân bằng giữa khai thác cục bộ (exploitation) và khám phá toàn cục (exploration).

Mặc dù ABC có ưu điểm về sự đơn giản và ít tham số điều khiển, thuật toán vẫn gặp hạn chế về tốc độ hội tụ chậm trong giai đoạn tìm kiếm tinh chỉnh. Do đó, báo cáo mở rộng phân tích các biến thể cải tiến trọng tâm là GABC (Gbest-guided ABC) và qABC (Quick ABC), nhằm khắc phục nhược điểm này thông qua cơ chế dẫn hướng và chọn lọc lân cận. Ngoài ra, báo cáo còn so sánh hiệu năng giữa thuật toán ABC gốc và hai biến thể, đồng thời đề cập đến các ứng dụng trong bài toán truyền thông, bao gồm tối ưu hóa tài nguyên mạng, ước lượng kênh truyền MIMO/OFDM và định tuyến trong mạng cảm biến không dây (WSN).

Từ khóa: Artificial Bee Colony, ABC, GABC, qABC, Tối ưu hóa bầy đàn, Mạng viễn thông, MIMO/OFDM.

1 Giới thiệu chung

1.1 Bối cảnh

Trong kỷ nguyên công nghệ thông tin và chuyển đổi số hiện nay, các bài toán tối ưu hóa đóng vai trò then chốt trong việc nâng cao hiệu suất của các hệ thống kỹ thuật, từ quy hoạch mạng lưới, phân bổ tài nguyên đến xử lý tín hiệu. Các phương pháp tối ưu hóa truyền thống dựa trên toán học (như đạo hàm, quy hoạch tuyến tính) thường gặp khó khăn hoặc trở nên kém hiệu quả khi đối mặt với các bài toán có không gian tìm kiếm lớn, hàm mục tiêu phức tạp, phi tuyến tính hoặc không khả vi (non-differentiable).

Để giải quyết thách thức này, các thuật toán Metaheuristic dựa trên trí tuệ bầy đàn (Swarm Intelligence) đã ra đời và phát triển mạnh mẽ. Nổi bật trong số đó là thuật toán Bầy ong nhân tạo (Artificial Bee Colony - ABC) do Karaboga đề xuất năm 2005 [?]. Với cơ chế mô phỏng hành vi tìm kiếm thức ăn thông minh của loài ong mật, ABC đã chứng minh được sự ưu việt về tính đơn giản, ít tham số điều khiển và khả năng tìm kiếm toàn cục mạnh mẽ.

Tuy nhiên, như nhiều thuật toán ngẫu nhiên khác, ABC vẫn tồn tại những hạn chế nhất định, điển hình là tốc độ hội tụ chậm ở giai đoạn cuối và khả năng khai thác cục bộ chưa tối ưu. Trong bối cảnh ngành Kỹ thuật Viễn thông (Telecommunications) đang đòi hỏi độ chính xác và tốc độ xử lý ngày càng cao (như trong mạng 5G/6G, IoT), việc nghiên cứu sâu về ABC, khắc phục các nhược điểm của nó thông qua các biến thể cải tiến và ứng dụng vào thực tiễn là một hướng đi cấp thiết và có ý nghĩa khoa học lớn.

1.2 Mục tiêu nghiên cứu

Dựa trên bối cảnh nêu trên, báo cáo này được thực hiện với các mục tiêu chính sau:

- **Nghiên cứu lý thuyết nền tảng:** Tìm hiểu sâu về nguồn gốc, cơ chế sinh học và mô hình toán học của thuật toán ABC để nắm bắt nguyên lý hoạt động cốt lõi.
- **Phân tích và Đánh giá:** Xác định rõ các ưu điểm và đặc biệt là các hạn chế của thuật toán ABC gốc (như vấn đề cân bằng giữa quá trình khai thác và khám phá).
- **Khảo sát các biến thể cải tiến:** Tập trung phân tích các biến thể hiện đại như GABC (Gbest-guided ABC) và qABC (Quick ABC) để hiểu cách thức khắc phục nhược điểm về tốc độ hội tụ và độ chính xác.
- **Ứng dụng thực tiễn:** Vận dụng kiến thức đã tìm hiểu để giải quyết các bài toán cụ thể trong lĩnh vực Kỹ thuật Viễn thông như: tối ưu hóa tài nguyên mạng, ước lượng kênh truyền MIMO/OFDM và định tuyến trong mạng cảm biến không dây (WSN).

1.3 Cấu trúc báo cáo

Để giải quyết trọn vẹn các mục tiêu nghiên cứu đã đề ra, bố cục của báo cáo được tổ chức thành 6 chương với trình tự phát triển từ lý thuyết nền tảng đến kiểm chứng và ứng dụng thực tiễn.

- **Chương 1:** Giới thiệu tổng quan về bối cảnh, tính cấp thiết và phạm vi của đề tài.
- **Chương 2:** Đi sâu vào nguyên lý hoạt động sinh học và mô hình toán học của thuật toán ABC gốc.
- **Chương 3:** Phân tích các hạn chế của thuật toán truyền thống và trình bày chi tiết hai biến thể cải tiến là GABC và qABC.

- **Chương 4:** Trình bày quá trình thiết lập môi trường mô phỏng và đánh giá hiệu năng hội tụ trên các hàm chuẩn.
- **Chương 5:** Vận dụng thuật toán vào giải quyết các bài toán cụ thể trong kỹ thuật truyền thông.
- **Chương 6:** Tổng kết kết quả, nhìn nhận hạn chế và đề xuất hướng phát triển.

2 Thuật toán Artificial Bee Colony (ABC)

2.1 Nguồn gốc và lịch sử

Thuật toán Bầy ong nhân tạo (Artificial Bee Colony - ABC) là một phương pháp tối ưu hóa ngẫu nhiên (stochastic optimization) thuộc lớp các thuật toán Metaheuristic. Thuật toán này được đề xuất lần đầu tiên bởi Dervis Karaboga (Đại học Erciyes, Thổ Nhĩ Kỳ) vào năm 2005 thông qua báo cáo kỹ thuật số TR06. Sau đó, nền tảng lý thuyết và hiệu năng của thuật toán tiếp tục được Karaboga và Basturk củng cố và công bố rộng rãi trên tạp chí *Journal of Global Optimization* vào năm 2007.

ABC ra đời dựa trên nguồn cảm hứng từ hành vi tìm kiếm thức ăn thông minh (foraging behavior) của đàn ong mật. Sự thành công của đàn ong trong tự nhiên không đến từ trí tuệ của một cá thể đơn lẻ, mà xuất phát từ "Trí tuệ bầy đàn" (Swarm Intelligence) – khả năng tự tổ chức và tương tác tập thể. Trong bản đồ khoa học máy tính, ABC đứng cùng hàng ngũ với các thuật toán nổi tiếng khác như Tối ưu hóa bầy đàn hạt (PSO) hay Tối ưu hóa đàn kiến (ACO), nhưng được đánh giá cao nhờ cấu trúc đơn giản và khả năng khám phá mạnh mẽ.

2.2 Phân loại và đặc tính thuật toán

2.2.1 Phân loại

Có thể phân loại thuật toán dựa trên các đặc tính kỹ thuật sau:

- **Metaheuristic (Siêu thuật toán):** ABC không sử dụng đạo hàm hay gradient của hàm mục tiêu, do đó nó có thể áp dụng cho các bài toán mà hàm mục tiêu không liên tục, không khả vi hoặc có độ phức tạp cao. Thuật toán không cam kết tìm ra nghiệm tối ưu tuyệt đối (global optimum) trong mọi trường hợp, nhưng đảm bảo tìm được nghiệm "đủ tốt" (near-optimal) với chi phí tính toán chấp nhận được.
- **Trí tuệ bầy đàn (Swarm Intelligence):** Hệ thống vận hành phi tập trung. Không có một "trung tâm điều khiển" nào chỉ đạo từng con ong phải bay đi đâu. Thay vào đó, hành vi của toàn hệ thống nảy sinh từ sự tương tác cục bộ giữa các cá thể (ong thợ, ong quan sát, ong trinh sát) và môi trường.
- **Tối ưu hóa dựa trên quần thể (Population-based):** ABC duy trì và cải thiện một tập hợp các nghiệm (quần thể) song song qua các thế hệ (vòng lặp), thay vì chỉ phát triển một nghiệm duy nhất như thuật toán tối luyện mô phỏng (Simulated Annealing).

2.2.2 Ưu điểm và Nhược điểm

Bảng 1: Tóm tắt ưu nhược điểm của thuật toán ABC

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> - Đơn giản và Linh hoạt: Dễ dàng cài đặt, chỉ dùng các phép toán số học cơ bản. - Ít tham số điều khiển: Chỉ cần chỉnh SN, MCN và $limit$. - Khả năng Khám phá tốt: Cơ chế ong trinh sát giúp thoát khỏi bẫy cực trị địa phương. 	<ul style="list-style-type: none"> - Tốc độ hội tụ chậm: Khả năng tinh chỉnh nghiệm (exploitation) ở giai đoạn cuối thường chậm hơn các phương pháp gradient. - Hạn chế trên không gian rời rạc: Thiết kế gốc cho miền liên tục, cần lai ghép để giải bài toán tổ hợp.

2.3 Nguyên lý hoạt động của đàn ong mật (Cơ chế tự nhiên)

Mô hình sinh học của ABC dựa trên ba thành phần chính và cơ chế giao tiếp thông minh:

2.3.1 Ba loại ong chức năng

1. **Ong thợ (Employed Bees):** Mỗi ong thợ gắn liền với một nguồn thức ăn cụ thể. Nhiệm vụ của chúng là khai thác nguồn thức ăn đó, ghi nhớ vị trí và lượng mật, sau đó quay về tổ để chia sẻ thông tin. Số lượng ong thợ thường bằng số lượng nguồn thức ăn (một nửa quần thể).
2. **Ong quan sát (Onlooker Bees):** Những con ong này đợi ở tổ (khu vực sàn nhảy). Chúng không tự đi tìm kiếm ban đầu mà chờ nhận thông tin từ ong thợ, sau đó sử dụng thông tin này để quyết định chọn nguồn thức ăn nào hứa hẹn nhất để bay đến khai thác.
3. **Ong trinh sát (Scout Bees):** Là những con ong thợ bị "thất nghiệp" do nguồn thức ăn của chúng đã cạn kiệt. Chúng bay đi tìm kiếm ngẫu nhiên các nguồn thức ăn mới trong môi trường để duy trì sự đa dạng nguồn sống cho đàn.

2.3.2 Cơ chế chia sẻ thông tin (Waggle Dance)

Trong tự nhiên, ong thợ giao tiếp bằng "điệu múa lắc" (Waggle Dance). Góc của điệu nhảy chỉ hướng bay; thời lượng và cường độ rung lắc tỷ lệ thuận với chất lượng (lượng mật) của nguồn thức ăn. Ong quan sát sẽ có xu hướng chọn những nguồn thức ăn được "quảng cáo" mạnh mẽ nhất.



Hình 1: Minh họa điệu múa lắc (Waggle Dance) của ong thợ

2.3.3 Ánh xạ từ Tự nhiên sang Thuật toán

Hệ thống sinh học	Hệ thống tính toán (ABC)
Nguồn thức ăn	Nghiệm của bài toán (Solution vector)
Lượng mật hoa	Độ thích nghi (Fitness value)
Tốc độ khai thác	Số lần lặp (Iterations)
Ong thợ / Ong quan sát	Toán tử tìm kiếm cục bộ (Local Search/Exploitation)
Ong trinh sát	Toán tử tìm kiếm toàn cục (Global Search/Exploration)

Bảng 2: Sự tương quan giữa hệ thống sinh học và thuật toán ABC

2.4 Mô hình toán học của thuật toán ABC

Xét bài toán tối ưu hóa với D biến số (chiều không gian), quần thể gồm SN nguồn thức ăn. Vị trí của nguồn thức ăn thứ i được biểu diễn là vector $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$.

2.4.1 Công thức tìm kiếm lân cận (Sinh nghiệm mới)

Cả ong thợ và ong quan sát đều sử dụng công thức này để tìm kiếm một vị trí mới v_{ij} xung quanh vị trí hiện tại x_{ij} :

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (1)$$

Trong đó:

- $k \in \{1, \dots, SN\}$: Chỉ số của một nghiệm ngẫu nhiên khác trong quần thể ($k \neq i$).
- $j \in \{1, \dots, D\}$: Một chiều không gian ngẫu nhiên được chọn để biến đổi.
- ϕ_{ij} : Một số thực ngẫu nhiên trong khoảng $[-1, 1]$, quyết định độ lớn và hướng bước nhảy.

2.4.2 Hàm đánh giá Fitness

Giá trị hàm mục tiêu $f(x_i)$ của bài toán (thường là cực tiểu hóa) được chuyển đổi sang giá trị độ thích nghi fit_i :

$$fit_i = \begin{cases} \frac{1}{1 + f(x_i)} & \text{nếu } f(x_i) \geq 0 \\ 1 + |f(x_i)| & \text{nếu } f(x_i) < 0 \end{cases} \quad (2)$$

2.4.3 Công thức tính xác suất chọn lựa

Ông quan sát chọn nguồn thức ăn i dựa trên xác suất p_i , tỷ lệ với độ thích nghi (cơ chế Roulette Wheel):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (3)$$

2.5 Mã giả và quy trình hoạt động

Algorithm 1: Thuật toán Bầy ong nhân tạo (ABC)

```
Input : Số lượng nguồn thức ăn ( $SN$ ), Giới hạn thử ( $limit$ ), Số vòng lặp tối đa ( $MCN$ )
Output: Nghiệm tối ưu toàn cục ( $x_{best}$ )
/* 1. Giai đoạn Khởi tạo */
1 Khởi tạo quần thể  $x_i$  ( $i = 1...SN$ ) ngẫu nhiên trong miền giá trị
2 Tính toán  $f(x_i)$  và độ thích nghi  $fit_i$ 
3 Thiết lập biến đếm  $trial\_counter_i = 0, \forall i$ 
4  $iter \leftarrow 0$ 
5 while  $iter < MCN$  do
    /* 2. Pha Ong Thợ (Employed Bees) */
    6 for  $i \leftarrow 1$  to  $SN$  do
        7 Sinh nghiệm mới  $v_i$  theo công thức (1)
        8 Tính  $fit(v_i)$ 
        9 if  $fit(v_i) > fit(x_i)$  then
            10  $x_i \leftarrow v_i$ 
            11  $trial\_counter_i \leftarrow 0$ 
        12 else
            13  $trial\_counter_i \leftarrow trial\_counter_i + 1$ 
    /* 3. Pha Ong Quan Sát (Onlooker Bees) */
    14 Tính xác suất  $p_i$  theo công thức (3)
    15  $t \leftarrow 0, i \leftarrow 1$ 
    16 while  $t < SN$  do
        17 if  $rand(0, 1) < p_i$  then
            18 Sinh nghiệm mới  $v_i$  từ  $x_i$ 
            19 Tính  $fit(v_i)$ 
            20 if  $fit(v_i) > fit(x_i)$  then
                21  $x_i \leftarrow v_i$ 
                22  $trial\_counter_i \leftarrow 0$ 
            23 else
                24  $trial\_counter_i \leftarrow trial\_counter_i + 1$ 
        25  $t \leftarrow t + 1$ 
    26  $i \leftarrow (i \bmod SN) + 1$ 
    /* 4. Pha Ong Trinh Sát (Scout Bees) */
    27 Tìm  $x_i$  có  $trial\_counter_i > limit$ 
    28 if Tồn tại  $x_i$  then
        29 Khởi tạo lại  $x_i$  ngẫu nhiên:  $x_i = x_{min} + rand(0, 1)(x_{max} - x_{min})$ 
        30  $trial\_counter_i \leftarrow 0$ 
    31 Cập nhật nghiệm tốt nhất toàn cục  $x_{best}$ 
    32  $iter \leftarrow iter + 1$ 
33 return  $x_{best}$ 
```

Thuật toán vận hành theo chu trình khép kín gồm 4 giai đoạn chính (Ong Thợ, Ong Quan Sát, Ong Trinh Sát) và lặp lại cho đến khi đạt điều kiện dừng (MCN vòng lặp hoặc sai số hội tụ).

3 Các biến thể của thuật toán ABC

Mặc dù thuật toán ABC cơ bản có ưu điểm về khả năng khám phá (Exploration) nhờ cơ chế của ong trinh sát, nhưng nó thường gặp hạn chế về tốc độ hội tụ chậm trong giai đoạn khai thác (Exploitation) do cơ chế tìm kiếm ngẫu nhiên. Để khắc phục, hai biến thể nổi bật là GABC và qABC đã được đề xuất.

3.1 Gbest-guided Artificial Bee Colony (GABC)

3.1.1 Ý tưởng chủ đạo

GABC, được đề xuất bởi Zhu và Kwong (2010), lấy cảm hứng từ thuật toán Tối ưu hóa bầy đàn (PSO). Ý tưởng cốt lõi là tích hợp thông tin của nghiệm tốt nhất toàn cục (Global Best - x_{best}) vào phương trình tìm kiếm để định hướng các con ong di chuyển về phía vùng tiềm năng nhất.

3.1.2 Phương trình cải tiến

Phương trình sinh nghiệm mới (1) của ABC gốc được sửa đổi thành:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \quad (4)$$

Trong đó:

- y_j : Là tham số thứ j của nghiệm tốt nhất toàn cục (x_{best}) tìm được hiện tại.
- ψ_{ij} : Là một số ngẫu nhiên trong khoảng $[0, C]$, với C là hệ số điều chỉnh lực hút (thường chọn $C = 1.5$).

Thành phần thứ ba $\psi_{ij}(y_j - x_{ij})$ đóng vai trò gia tăng khả năng khai thác, giúp thuật toán hội tụ nhanh hơn.

3.1.3 Phân tích sự khác biệt về cơ chế hoạt động

Sự khác biệt cốt lõi giữa ABC gốc và GABC nằm ở vectơ chỉ hướng di chuyển của các cá thể ong:

1. **Cơ chế ABC gốc (Chuyển động Brown ngẫu nhiên):** Con ong i di chuyển về phía (hoặc ra xa) một con ong k được chọn ngẫu nhiên. Vì k có thể là một nghiệm tốt hoặc xấu, bước nhảy này mang tính chất “mò mẫm”, giúp duy trì sự đa dạng sinh học cao nhưng tốn nhiều thời gian để tinh chỉnh nghiệm chính xác.
2. **Cơ chế GABC (Chuyển động có định hướng):** Phương trình (4) tạo ra một lực đẩy tổng hợp từ hai phía:
 - Lực ngẫu nhiên từ x_{kj} : Giúp duy trì khả năng khám phá, tránh bị mắc kẹt sớm.
 - Lực hút từ x_{best} : Kéo nghiệm hiện tại di chuyển nhanh hơn về phía vùng hứa hẹn nhất của không gian tìm kiếm.

Nhờ sự kết hợp này, GABC không loại bỏ hoàn toàn tính ngẫu nhiên (tránh hội tụ sớm) nhưng gia tăng đáng kể áp lực chọn lọc, buộc các con ong phải tập trung khai thác xung quanh nghiệm tốt nhất.

Algorithm 2: Gbest-guided Artificial Bee Colony (GABC)

Input : Số lượng nguồn thức ăn SN , Giới hạn thử $limit$, Số vòng lặp tối đa MCN
Output: Nghiệm tối ưu x_{best}

```

/* 1. Khởi tạo */
1 Khởi tạo quần thể  $x_i$  ngẫu nhiên; tính  $fit_i$ ;  $trial_i \leftarrow 0$ ;  $iter \leftarrow 0$ 
2 while  $iter < MCN$  do
    /* 2. Pha Ong Thợ */
    3 for  $i \leftarrow 1$  to  $SN$  do
        4 Sinh nghiệm mới:  $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(x_{best,j} - x_{ij})$ 
        5 Cập nhật nếu  $fit(v_i) > fit(x_i)$ , ngược lại  $trial_i \leftarrow trial_i + 1$ 
    /* 3. Pha Ong Quan Sát */
    6 Tính xác suất  $p_i$ ; chọn và cập nhật  $v_i$  theo  $x_i +$  hướng  $x_{best}$ 
    /* 4. Pha Ong Trinh Sát */
    7 Thay  $x_i$  nếu  $trial_i > limit$ 
    8 Cập nhật  $x_{best}$ ;  $iter \leftarrow iter + 1$ 
9 return  $x_{best}$ 

```

3.1.4 Vai trò quan trọng của tham số điều chỉnh C

Hệ số C trong khoảng của $\psi_{ij} \in [0, C]$ đóng vai trò then chốt trong việc cân bằng giữa Khám phá (Exploration) và Khai thác (Exploitation):

- **Trường hợp $C = 0$:** Thành phần Gbest bị triệt tiêu ($\psi_{ij} = 0$). GABC trở về trạng thái của thuật toán ABC nguyên bản.
- **Trường hợp C nhỏ (ví dụ $C < 1$):** Lực hút về phía x_{best} yếu. Thuật toán vẫn ưu tiên tìm kiếm diện rộng, tốc độ hội tụ được cải thiện nhẹ nhưng an toàn trước các cực trị địa phương.
- **Trường hợp C lớn (ví dụ $C > 2$):** Lực hút về phía x_{best} rất mạnh. Các con ong sẽ lao nhanh về phía nghiệm tốt nhất. Điều này giúp tốc độ hội tụ cực nhanh (thích hợp cho các hàm đơn cực - unimodal), nhưng làm tăng rủi ro hội tụ sớm (premature convergence) tại các cực trị địa phương đối với các hàm đa cực phức tạp.

Theo các nghiên cứu thực nghiệm của Zhu và Kwong, giá trị $C = 1.5$ thường mang lại sự cân bằng tốt nhất cho đa số các bài toán tối ưu hóa số học.

3.2 Quick Artificial Bee Colony (qABC)**3.2.1 Nguồn gốc và Cảm hứng sinh học**

Biến thể qABC được đề xuất bởi Dervis Karaboga và Beyza Gorkemli vào năm 2014 – chính nhóm tác giả cha đẻ của thuật toán ABC gốc. Động lực phát triển qABC xuất phát từ việc quan sát sâu hơn vào hành vi xã hội thực tế của loài ong mật.

Trong tự nhiên, một con ong quan sát (Onlooker bee) tại khu vực nhảy (dance floor) không có khả năng quan sát toàn bộ các điệu nhảy của cả đàn cùng một lúc. Thay vào đó, nó chịu ảnh hưởng mạnh mẽ nhất từ những con ong thợ đang nhảy ở vị trí vật lý gần nó nhất (lân cận). qABC mô phỏng hành vi “tương tác cục bộ” này để cải thiện hiệu suất, thay vì giả định tương tác toàn cục như thuật toán gốc.

3.2.2 Cơ chế “Hàng xóm tinh anh” (Elite Neighbor Strategy)

Sự khác biệt căn bản giữa qABC và ABC gốc nằm ở cách thức Ong Quan Sát lựa chọn thông tin để khai thác. Quy trình này được cụ thể hóa qua các bước toán học sau:

Bước 1: Xác định vùng lân cận (Neighborhood Definition)

Với mỗi nguồn thức ăn x_i được chọn, thuật toán xác định một tập hợp các “hàng xóm” N_i dựa trên khoảng cách Euclid. Một nguồn thức ăn x_k được coi là hàng xóm của x_i nếu:

$$d(x_i, x_k) \leq r \times d_{max} \quad (5)$$

Trong đó:

- $d(x_i, x_k)$: Khoảng cách Euclid giữa hai vector nghiệm.

- d_{max} : Khoảng cách lớn nhất có thể trong không gian tìm kiếm.
- r : Bán kính lân cận (Neighborhood Radius), $r \in (0, 1]$.

Bước 2: Chọn lọc tinh anh (Elite Selection)

Thay vì chọn một đối tác k ngẫu nhiên bất kỳ như ABC gốc, qABC tìm kiếm nghiệm tốt nhất trong tập hàng xóm N_i vừa xác định:

$$x_{best}^N = \arg \max_{k \in N_i} \{fit(x_k)\} \quad (6)$$

Nghiệm này được gọi là “Hàng xóm tinh anh” (Elite Neighbor).

Bước 3: Phương trình sinh nghiệm mới

Đây là điểm khác biệt lớn nhất về mặt toán học. Ong quan sát sẽ dịch chuyển tâm tìm kiếm từ vị trí hiện tại x_i sang vị trí của hàng xóm tinh anh x_{best}^N để tận dụng triệt để chất lượng của vùng nghiệm tốt:

$$v_{ij} = x_{best,j}^N + \phi_{ij}(x_{best,j}^N - x_{kj}) \quad (7)$$

Algorithm 3: Quick Artificial Bee Colony (qABC)

Input : Số lượng nguồn thức ăn SN , Giới hạn thử $limit$, Số vòng lặp tối đa MCN , Bán kính lân cận r

Output: Nghiệm tối ưu x_{best}

/* 1. Khởi tạo */

1 Khởi tạo quần thể x_i ; tính fit_i ; $trial_i \leftarrow 0$; $iter \leftarrow 0$

2 **while** $iter < MCN$ **do**

 /* 2. Pha Ong Thợ (giống ABC) */

3 **for** $i \leftarrow 1$ **to** SN **do**

 Sinh nghiệm mới v_i theo công thức chuẩn ABC

5 **if** $fit(v_i) > fit(x_i)$, ngược lại $trial_i \leftarrow trial_i + 1$

 /* 3. Pha Ong Quan Sát (Onlooker) với neighbourhood-search */

6 **for each onlooker bee do**

 Chọn một nguồn thức ăn x_i theo xác suất roulette dựa trên fit_i

8 Xác định tập lân cận $N_i = \{x_m \mid dist(x_m, x_i) \leq r\}$

9 Tìm $x_{nbest} = \arg \max_{x \in N_i \cup \{x_i\}} fit(x)$

10 Sinh nghiệm mới v_i quanh x_{nbest} :

$$v_{ij} = x_{best,j}^N + \phi_{ij}(x_{best,j}^N - x_{kj})$$

 (với x_k ngẫu nhiên, $k \neq i$)

11 **if** $fit(v_i) > fit(x_i)$ thì $x_i \leftarrow v_i$, $trial_i \leftarrow 0$, else $trial_i \leftarrow trial_i + 1$

 /* 4. Pha Ong Trinh Sát (Scout) */

12 Với mọi i có $trial_i > limit$, khởi tạo mới x_i

13 Cập nhật x_{best} ; $iter \leftarrow iter + 1$

14 **return** x_{best}

So sánh với ABC gốc:

- *ABC gốc*: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$. Tâm tìm kiếm là chính nó, hướng đi là ngẫu nhiên.
- *qABC*: $v_{ij} = x_{best,j}^N + \phi_{ij}(x_{best,j}^N - x_{kj})$. Tâm tìm kiếm là nghiệm tốt nhất lân cận. Điều này tạo ra khả năng tinh chỉnh (exploitation) cục bộ cực mạnh.

3.2.3 Ảnh hưởng của Bán kính lân cận (r)

Tham số r đóng vai trò như một “ống kính” điều chỉnh tầm nhìn của ong, quyết định sự cân bằng của thuật toán:

1. **Khi r nhỏ (Lân cận hẹp)**: Tập hàng xóm N_i chỉ chứa vài cá thể rất gần nhau. Thuật toán tập trung đào sâu vào các cực trị địa phương (Local Exploitation). Điều này giúp độ chính xác cực cao nhưng có nguy cơ mắc kẹt nếu bán kính quá nhỏ khiến ong không nhìn thấy vùng tốt hơn ở xa.
2. **Khi r lớn (Lân cận rộng)**: Tập hàng xóm mở rộng ra toàn đàn. Khi $r = 1$, x_{best}^N chính là nghiệm tốt nhất toàn cục ($gbest$). Lúc này, hành vi của qABC sẽ tiệm cận với GABC, tăng khả năng hội tụ nhanh nhưng giảm tính đa dạng cục bộ.

Kết quả thực nghiệm cho thấy qABC vượt trội hơn ABC gốc và GABC trên các hàm mục tiêu phức tạp (đa cực) nhờ khả năng giữ cân bằng tốt giữa việc khai thác vùng tốt nhất trong tầm nhìn (nhờ x_{best}^N) và duy trì sự đa dạng của các nhóm lân cận khác nhau.

4 Mô phỏng và đánh giá hiệu năng

Mã nguồn của các thuật toán được cài đặt và kiểm chứng trên nền tảng **Google Colab**¹.

Do bản chất ngẫu nhiên (stochastic nature) của các thuật toán tối ưu hóa bầy đàn, kết quả của một lần chạy đơn lẻ có thể bị ảnh hưởng bởi việc khởi tạo quần thể ban đầu và không phản ánh chính xác hiệu năng tổng thể. Để đảm bảo tính khách quan và độ tin cậy thống kê, chúng tôi áp dụng quy trình đánh giá như sau:

1. Mỗi thuật toán (ABC, GABC, qABC) được thực thi độc lập **30 lần** ($N_{runs} = 30$) trên cùng một hàm mục tiêu.
2. Tại mỗi vòng lặp, giá trị độ thích nghi tốt nhất (Best Fitness) của quần thể được ghi lại.
3. Kết quả cuối cùng được biểu diễn dưới dạng **giá trị trung bình** (Average Fitness) của 30 lần chạy để loại bỏ các yếu tố ngẫu nhiên và so sánh tốc độ hội tụ thực tế.

4.1 Thí nghiệm 1: Đánh giá tốc độ hội tụ trên hàm Sphere

4.1.1 Thiết lập bài toán

- **Mục đích:** Kiểm tra khả năng **Khai thác (Exploitation)** – tức là tốc độ hội tụ của thuật toán trong một không gian tìm kiếm đơn cực (unimodal) và trơn (convex). Đây là bài toán cơ bản để đo xem thuật toán nào tinh chỉnh nghiệm nhanh nhất.
- **Mô tả hàm:** Sphere là hàm lồi dạng parabol, chỉ có duy nhất một cực trị toàn cục tại tâm.
- **Công thức hàm mục tiêu** ($D = 5$):

$$F_{sphere}(\mathbf{x}) = \sum_{i=1}^D x_i^2 \quad (8)$$

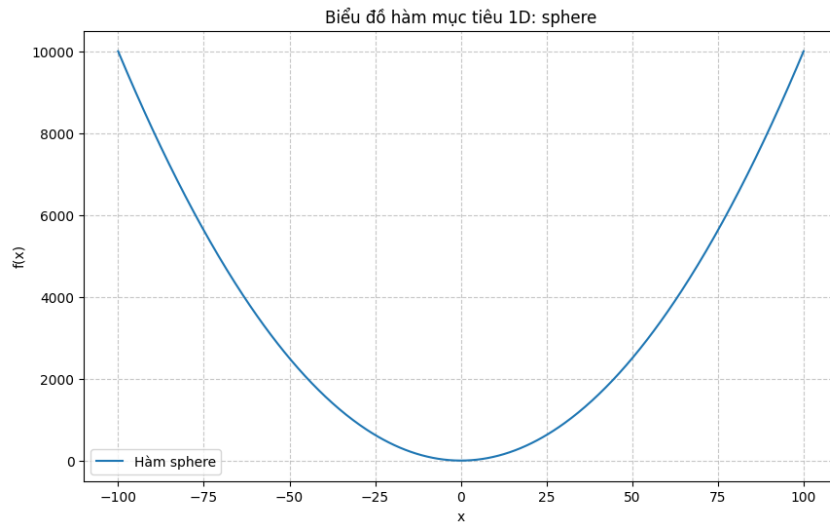
Trong đó: Biến số $x_i \in [-100, 100]$. Giá trị tối ưu toàn cục (Global Min) bằng 0 tại $\mathbf{x} = [0, 0, \dots, 0]$.

- **Tham số cấu hình:**
 - Kích thước quần thể (SN): 50 (25 Ong thợ, 25 Ong quan sát).
 - Số vòng lặp tối đa (MCN): 100.
 - Giới hạn thử ($limit$): 10.
 - Hệ số C (cho GABC): 1.5.
 - Bán kính lân cận r (cho qABC): 0.9.
 - Vùng tìm kiếm: $[-100, 100]$.

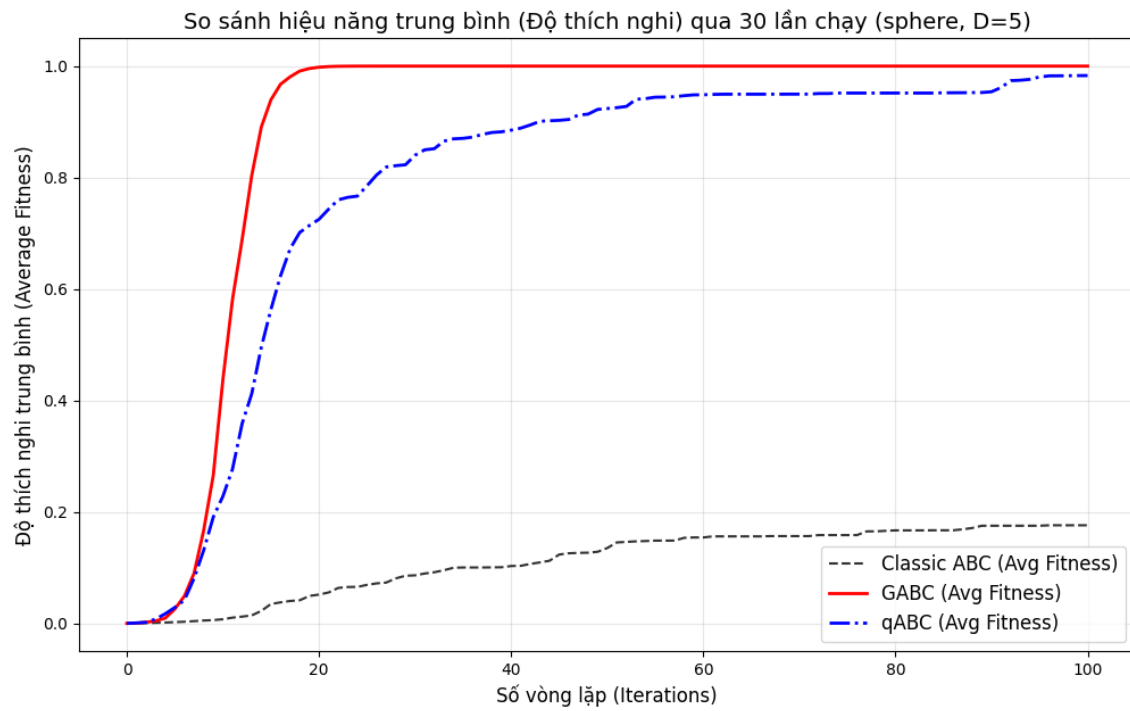
4.1.2 Kết quả và Thảo luận

Biểu đồ dưới đây minh họa quá trình tăng trưởng độ thích nghi và giảm thiểu sai số trung bình của ba thuật toán qua các vòng lặp.

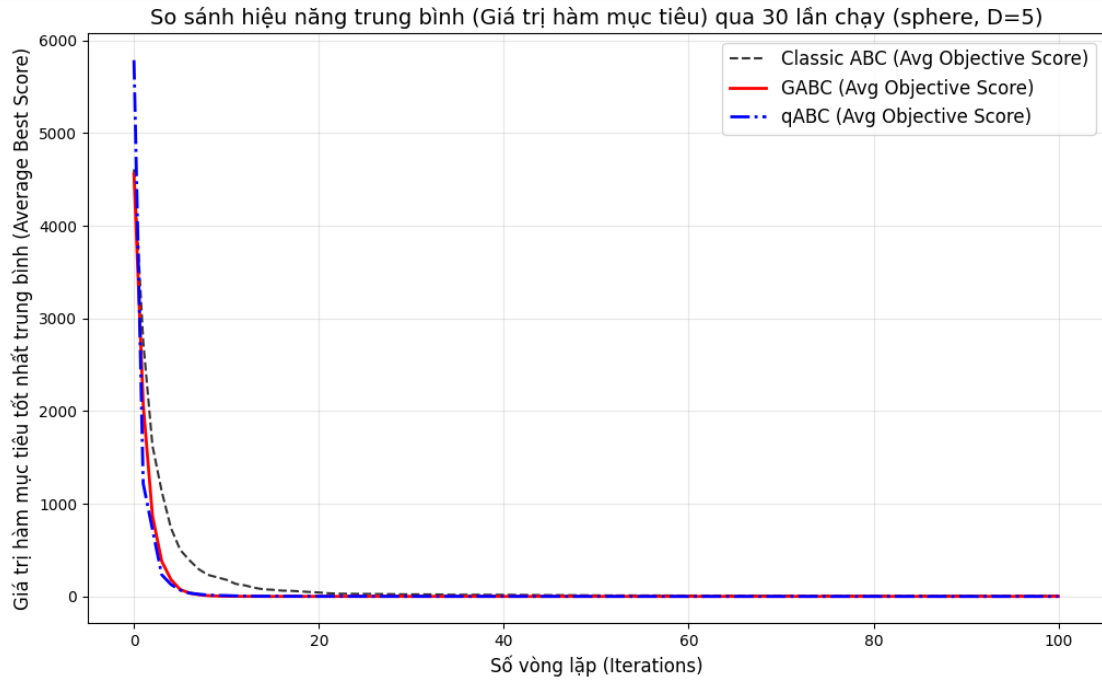
¹Mã nguồn chi tiết và kịch bản mô phỏng có tại: https://colab.research.google.com/drive/15v51eT78qKiPJkkHazGJ_gL-1Se0GkaF?usp=sharing



Hình 2: Đồ thị hàm mục tiêu Sphere (1D).



Hình 3: Biểu đồ so sánh hiệu năng trung bình (giá trị Fitness) trên hàm Sphere.



Hình 4: Biểu đồ so sánh hiệu năng trung bình (giá trị hàm mục tiêu - Score) trên hàm Sphere.

Bảng 3: Thống kê giá trị tối ưu trung bình trên hàm Sphere sau 30 lần chạy

Thuật toán	Giá trị trung bình (Mean Score)	Độ lệch chuẩn (Std Dev)
Classic ABC	6.60	4.21
qABC ($r \approx 1$)	0.021	0.071
GABC ($C = 1.5$)	4.98×10^{-17}	1.89×10^{-17}

Dựa trên biểu đồ hội tụ (Hình 3) và bảng số liệu thống kê (Bảng 3), chúng tôi rút ra các phân tích sau:

Phân tích kết quả:

- **GABC (Hiệu năng vượt trội):** Đạt kết quả trung bình là $\approx 5 \times 10^{-17}$ (một con số cực nhỏ, gần như bằng 0 tuyệt đối). Với độ lệch chuẩn cũng cực thấp, GABC cho thấy sự ổn định tuyệt vời. Điều này chứng minh cơ chế dẫn hướng toàn cục (*gbest*) hoạt động hoàn hảo trên các bề mặt trơn, kéo bầy ong lao thẳng xuống đáy mà không bị chệch hướng.
- **qABC (Hiệu quả cao):** Đạt kết quả **0.021**, tốt hơn gấp **300 lần** so với thuật toán gốc (6.60). Tuy không thể đạt độ chính xác “máy tính” như GABC, nhưng qABC vẫn chứng tỏ khả năng tinh chỉnh nghiệm cục bộ rất đáng nể nhờ cơ chế lân cận.
- **Classic ABC (Khá):** Dù vẫn hội tụ, nhưng tốc độ chậm hơn nhiều và dừng lại ở mức lỗi **6.60**. Sự tìm kiếm ngẫu nhiên không đủ mạnh để đẩy nghiệm về sát 0 trong cùng số lượng vòng lặp.

4.2 Thí nghiệm 2: Đánh giá tốc độ hội tụ trên hàm Shifted Rastrigin

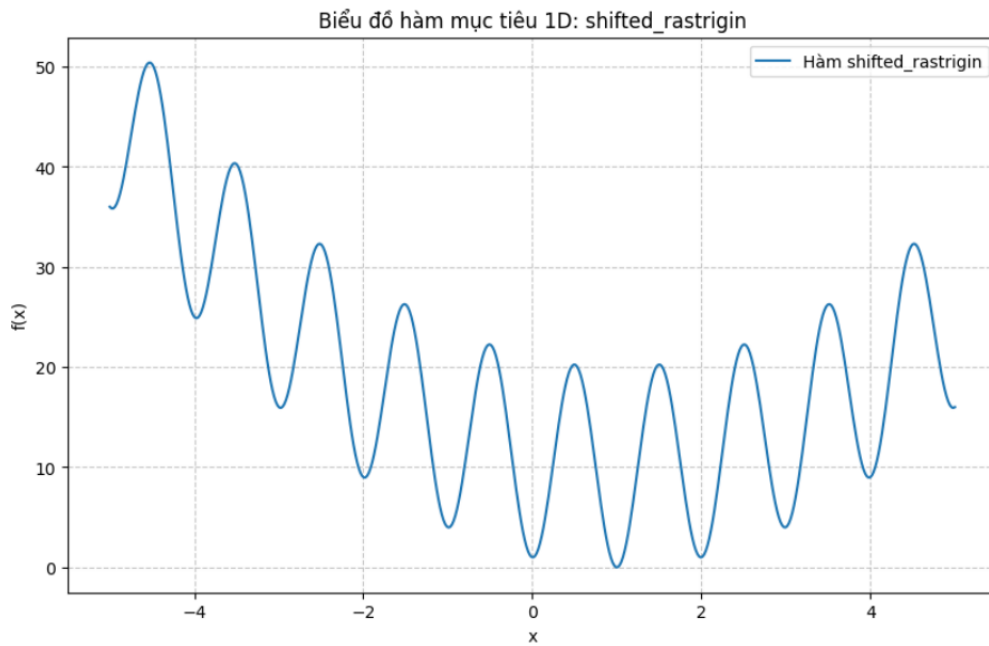
4.2.1 Thiết lập bài toán

- **Mục đích:** Kiểm tra khả năng **Khám phá (Exploration)** – tức là khả năng của thuật toán trong việc không bị kẹt vào các “ổ gà” (cực trị địa phương - local optima) trong không gian tìm kiếm đa chiều phức tạp ($D = 20$) để tiếp tục tìm kiếm nghiệm tối ưu toàn cục.
- **Mô tả hàm:** Hàm Rastrigin là một hàm phi tuyến điển hình có bề mặt lồi lõm rất phức tạp với hàng trăm cực trị địa phương phân bố dày đặc.
- **Công thức hàm mục tiêu ($D = 20$):**

$$F(\mathbf{x}) = \sum_{i=1}^D [(x_i - s)^2 - 10 \cos(2\pi(x_i - s)) + 10] \quad (9)$$

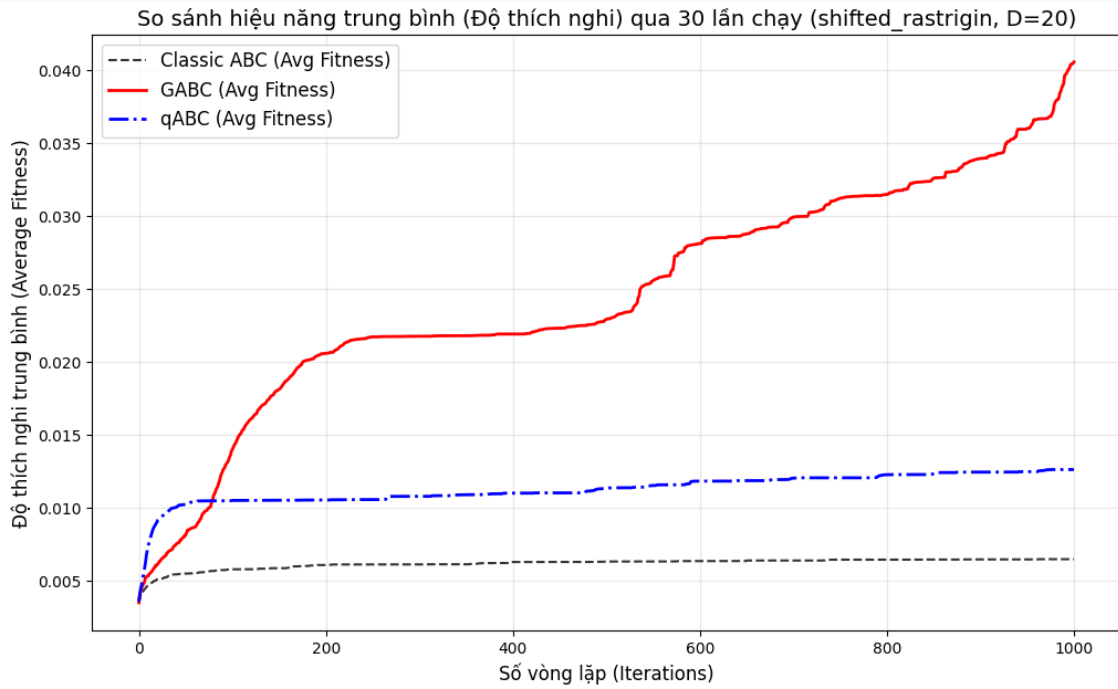
Trong đó:

- D : Số chiều không gian (Dimension).
- $\mathbf{x} = [x_1, x_2, \dots, x_D]$: Vector biến số, với $x_i \in [-5.0, 5.0]$.
- s : Giá trị dịch chuyển (Shift value), ví dụ $s = 1.0$. Khi đó cực trị toàn cục (Global Min) nằm tại $x_i = 1.0$ với giá trị $f(\mathbf{x}) = 0$.
- **Tham số cấu hình (Khác biệt so với TN1):**
 - Kích thước quần thể (SN): 50.
 - Số vòng lặp tối đa (MCN): **1000** (Tăng lên để quan sát khả năng thoát bẫy).
 - Vùng tìm kiếm: $[-5, 5]$.
 - Các tham số khác ($C, r, limit$) giữ nguyên.

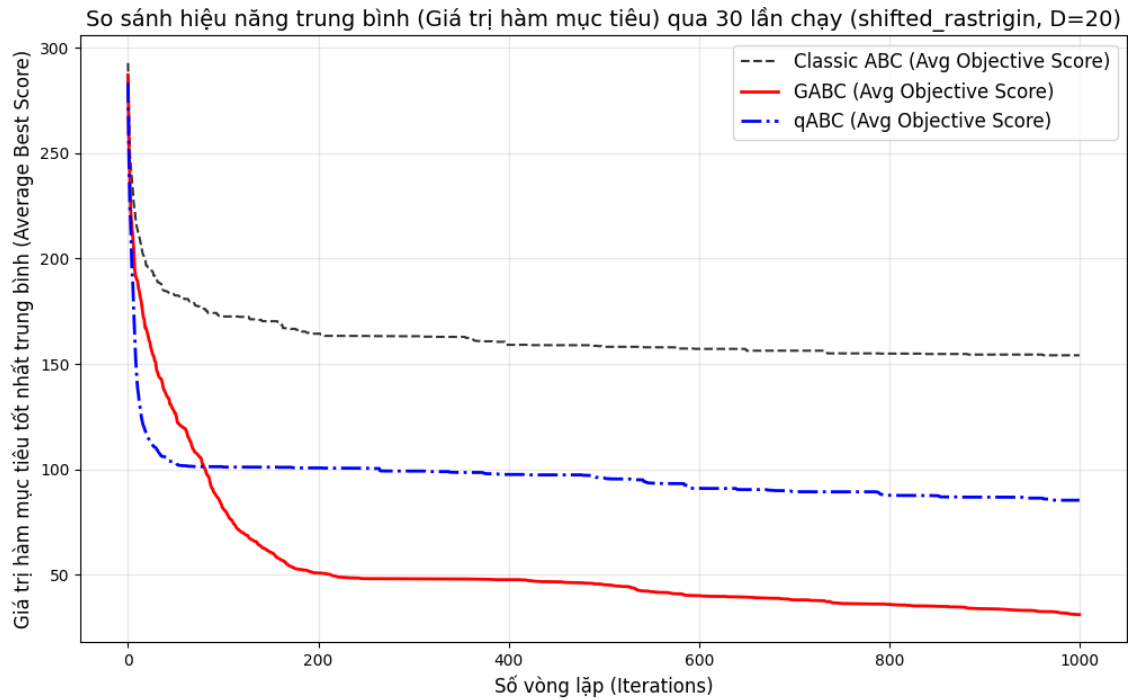


Hình 5: Đồ thị hàm mục tiêu Styblinski-Tang (1D).

4.2.2 Kết quả và Thảo luận



Hình 6: Biểu đồ so sánh hiệu năng trung bình (giá trị Fitness) trên hàm Rastrigin.



Hình 7: Biểu đồ so sánh giá trị hàm mục tiêu (Best Score) trên hàm Rastrigin.

Bảng 4: Thống kê giá trị hàm mục tiêu cuối cùng (Final Best Score) trên hàm Rastrigin

Thuật toán	Giá trị trung bình (Mean Score)	Đánh giá
Classic ABC	≈ 155.0	Kém (Bị kẹt sớm)
qABC ($r \approx 1$)	≈ 85.5	Khá (Thoát bẫy tốt)
GABC ($C = 1.5$)	≈ 30.2	Tốt nhất

Phân tích khả năng thoát bẫy (Trap Escape Capability):

- Khác với hàm Sphere có độ dốc rõ ràng, hàm Rastrigin với bề mặt lỗi lồi là thử thách lớn cho các thuật toán dựa trên gradient hoặc dẫn hướng quá mạnh. Dựa trên đồ thị Hình 7 và Bảng 4, ta có các quan sát sau:
- **Classic ABC (Đường màu đen):** Thể hiện hiệu năng kém nhất. Đường cong giảm nhanh ở 50 vòng lặp đầu nhưng sau đó đi ngang (bão hòa) ở mức lỗi rất cao (≈ 155). Điều này chứng tỏ với số chiều lớn ($D = 20$), cơ chế tìm kiếm ngẫu nhiên hoàn toàn bất lực trong việc thoát khỏi vô số các cực trị địa phương của hàm Rastrigin.
- **qABC (Đường màu xanh):** Cho thấy khả năng **thoát bẫy ban đầu cực tốt**. Trong khoảng 100 vòng lặp đầu tiên, qABC giảm lỗi cực nhanh xuống mức 100, vượt xa Classic ABC. Tuy nhiên, ở giai đoạn sau (từ vòng 200 trở đi), tốc độ cải thiện bị chậm lại, cho thấy thuật toán vẫn cần thêm động lực để tinh chỉnh nghiệm sâu hơn nữa.
- **GABC (Đường màu đỏ):** Là thuật toán **bền bỉ và hiệu quả nhất**. Mặc dù khởi đầu chậm hơn qABC một chút, nhưng GABC duy trì được đà giảm lỗi liên tục suốt 1000 vòng lặp. Nhờ lực hút về phía nghiệm tốt nhất toàn cục (g_{best}), GABC đã “đào” sâu xuống được mức lỗi thấp nhất (≈ 30), bỏ xa hai thuật toán còn lại.

Kết luận: Thí nghiệm này khẳng định trong các bài toán có số chiều lớn và độ phức tạp cao, việc sử dụng các biến thể có cơ chế dẫn hướng (như GABC) là yếu tố tiên quyết để đạt được độ chính xác cao.

5 ỨNG DỤNG TRONG KỸ THUẬT TRUYỀN THÔNG

Các đặc tính ưu việt của ABC và các biến thể được ứng dụng rộng rãi trong các bài toán tối ưu mạng phức tạp:

5.1 Tối ưu hóa tài nguyên mạng (Resource Allocation)

Trong mạng 5G/6G, bài toán phân bổ công suất và băng thông thường là phi tuyến và không lồi (non-convex). ABC được sử dụng để tìm cấu hình công suất phát tối ưu cho các người dùng nhằm tối đa hóa tổng thông lượng (Throughput) và hiệu quả năng lượng (Energy Efficiency), đồng thời thỏa mãn các ràng buộc về nhiễu giao thoa (Interference).

5.2 Ước lượng kênh truyền trong hệ thống MIMO

Đối với hệ thống MIMO quy mô lớn (Massive MIMO), việc ước lượng chính xác ma trận kênh truyền là thách thức lớn. ABC, đặc biệt là biến thể qABC, được áp dụng để tìm kiếm vector tham số kênh sao cho sai số bình phương trung bình (MSE) giữa tín hiệu nhận và tín hiệu pilot là nhỏ nhất, giúp giảm tỷ lệ lỗi bit (BER) hiệu quả hơn so với các phương pháp truyền thống như LS hay MMSE.

5.3 Định tuyến trong Mạng cảm biến không dây (WSN)

ABC được sử dụng để giải quyết bài toán phân cụm (Clustering) và chọn nút trưởng cụm (Cluster Head) nhằm cân bằng năng lượng tiêu thụ giữa các nút cảm biến, từ đó kéo dài tuổi thọ mạng. Hàm mục tiêu được thiết kế dựa trên năng lượng dư thừa và khoảng cách truyền dẫn.

6 KẾT LUẬN

Báo cáo đã trình bày cơ sở lý thuyết vững chắc của thuật toán Bầy ong nhân tạo (ABC) và phân tích sâu hai biến thể GABC, qABC. Kết quả mô phỏng trên hàm Styblinski-Tang cho thấy sự đánh đổi rõ rệt giữa tốc độ (GABC) và độ chính xác (qABC), đồng thời khẳng định sự vượt trội của các biến thể so với thuật toán gốc. Hướng phát triển tiếp theo sẽ tập trung vào việc áp dụng qABC vào bài toán phân bổ tài nguyên động trong mạng 5G.

Tài liệu

- [1] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” *Technical Report-tr06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [2] G. Zhu and S. Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization,” *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [3] D. Karaboga and B. Gorkemli, “A quick artificial bee colony (qABC) algorithm and its performance on optimization problems,” *Applied Soft Computing*, vol. 23, pp. 227–238, 2014.
- [4] 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.