

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

BÁO CÁO
TÌM HIỂU THUẬT TOÁN ABC
VÀ CÁC BIẾN THẾ
(TRỌNG TÂM: ABC CHUẨN, GABC VÀ qABC)

Giảng viên: Trịnh Văn Chiên

Người hướng dẫn: Nguyễn Quang Đông

Nhóm 91

Thành viên	MSSV
Đào Thái Hoàng	20235720
Trần Thu Phương	20235811

Hà Nội, 2025

Mục lục

I THUẬT TOÁN BÂY ONG NHÂN TẠO (ABC)	2
1 Đặt vấn đề	2
2 Mô tả bài toán	2
3 Cơ sở lý thuyết thuật toán ABC	2
3.1 Ý tưởng sinh học	2
3.2 Ánh xạ sang bài toán tối ưu	3
4 Mô hình hóa bài toán	3
4.1 Công thức tính độ thích nghi (Fitness)	3
4.2 Cơ chế tìm kiếm lân cận (Ong thợ và Ong quan sát)	3
4.3 Cơ chế chọn lọc (Ong quan sát)	4
4.4 Cơ chế sinh ngẫu nhiên (Ong trinh sát)	4
5 Mã giả thuật toán	5
6 Đánh giá Ưu và Nhược điểm	6
6.1 Ưu điểm	6
6.2 Nhược điểm	6
II CÁC BIẾN THỂ THUẬT TOÁN ABC	6
7 Giới thiệu chung về các biến thể ABC	6
8 Phân tích Biến thể GABC	6
8.1 Hạn chế của Thuật toán ABC Chuẩn	6
8.2 Cơ chế hoạt động của GABC	7
9 Phân tích Biến thể qABC (Quick ABC)	7
9.1 Ý tưởng cốt lõi	7
9.2 Phương trình tìm kiếm qABC	7
9.3 Công thức Hybrid GABC - qABC	8
9.4 So sánh trực quan	8
10 So sánh Tổng hợp các Biến thể	8
11 Kết luận	8

Phần I

THUẬT TOÁN BẦY ONG NHÂN TẠO (ABC)

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** (**A**rtificial **B**e **C**olony), 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.

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.

3 Cơ sở lý thuyết thuật toán ABC

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.

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.

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).

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 .

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.
- ϕ_{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 ($trial_i$).

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.

4.4 Cơ chế sinh ngẫu nhiên (Ong trinh 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 trinh sát và sinh ra một nghiệm mới ngẫu nhiên:

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

5 Mã giả thuật toán

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

Đầu vào: SN (số nguồn thức ăn), $limit$, $MaxCycle$, D (số chiều)

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

6 Đánh giá Ưu và Nhược điểm

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.

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.

Phần II CÁC BIẾN THỂ THUẬT TOÁN ABC

7 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.

8 Phân tích Biến thể GABC

8.1 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:

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

Trong đó:

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

Phân tích: Phương trình (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.

8.2 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:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \quad (2)$$

Trong đó:

- y_j : Là phần tử thứ j của **nghiệm tốt nhất toàn cục** (Global Best).
- ψ_{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.

9 Phân tích Biến thể qABC (Quick ABC)

9.1 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.

9.2 Phương trình tìm kiếm qABC

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

$$v_{ij} = x_{best_neighbor,j} + \phi_{ij}(x_{best_neighbor,j} - x_{ij}) \quad (3)$$

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.

9.3 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 cục bộ mạnh của qABC và khả năng định hướng toàn cục của GABC được đề xuất như sau:

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

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.

9.4 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).

10 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 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.	Rất nhanh.
Rủi ro	Tốn thời gian.	Cân bằng tốt.	Dễ bị mắc kẹt ở cực trị địa phương (do hội tụ quá sớm).

11 Kết luận

Việc nghiên cứu các biến thể như GABC và qABC cho thấy sự linh hoạt của thuật toán Bầy ong.

- Nếu bài toán cần độ chính xác cao và không quá gấp về thời gian: **GABC** là lựa chọn an toàn và hiệu quả nhất (với tham số $C = 1.5$).
- Nếu bài toán đơn giản và cần tìm nghiệm cực nhanh: **qABC** là lựa chọn phù hợp.

Sự kết hợp giữa các cơ chế này là nền tảng để phát triển các thuật toán lai ghép (Hybrid) mạnh mẽ hơn trong tương lai.

Tài liệu

- [1] Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7), 3166-3173.
- [2] Karaboga, D., & Gorkemli, B. (2012). A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. *Applied Soft Computing*, 23, 227-238.