

THỰC NGHIỆM THUẬT TOÁN TỐI ƯU BA TRÊN CÁC HÀM SỐ THỰC

EXPERIMENTAL SWARM ALGORITHM ON OPTIMAL FUNCTIONS

NGUYỄN PHƯƠNG HẠC(*), NGUYỄN THU NGUYỆT MINH(**) và TRÀ VĂN ĐÔNG(***)

TÓM TẮT: Bài viết mô tả Thuật toán tối ưu BA (Bees Algorithm) mới dựa trên nghiên cứu hành vi tổ chức tìm kiếm thức ăn của bầy ong và thực nghiệm Thuật toán tối ưu BA trên một tập hợp các bài toán tối ưu hàm số thực để kiểm chứng những hiệu quả nổi trội của nó so với các phương pháp trong giải quyết các bài toán tối ưu.

Từ khóa: thuật toán tối ưu BA (Bees Algorithm); thuật giải tối ưu; hàm số tối ưu.

ABSTRACT: The BA Optimization Algorithm (Bees Algorithm) is a new optimization algorithm that today is very interested in because of its outstanding effects compared to other methods in solving optimal problems. This article presents the method of optimizing bee colonies, the effectiveness of which is evaluated on real function optimization problems. Experimental results on real functions result in faster and more accurate results.

Key words: bees Algorithm (BA); optimal algorithm; optimal function.

1. ĐẶT VẤN ĐỀ

Việc tìm kiếm một giải thuật tối ưu hiệu quả phục vụ cho công việc lập trình các hoạt động kinh tế xã hội là nhu cầu cấp thiết nhằm rút ngắn thời gian, giảm chi phí... và tối đa hóa lợi nhuận. Một trong những cách tiếp cận có hiệu quả đối với bài toán tối ưu đó là Thuật toán tối ưu BA (Bees Algorithm - BA). Phương pháp tối ưu dựa trên hành vi tìm kiếm thức ăn của bầy ong là một phương pháp Meta heuristic (kỳ vọng) mới. Bài viết tập trung tìm hiểu Thuật toán tối ưu BA và thực nghiệm nhằm mục đích so sánh đánh giá hiệu quả của các thuật toán đó trên một tập hợp các bài toán tối ưu hàm số thực.

2. NỘI DUNG

Trong bài viết, chúng tôi trình bày một số thuật toán bầy ong cơ bản như BA (Bees Algorithm), giới thiệu tổng quan thuật giải tối ưu hóa bầy đàn; giới thiệu thuật giải tối ưu

BABA và bầy ong trong tự nhiên; trình bày mã giả các thuật toán bầy ong và tổng kết thuật toán bầy ong.

2.1. Tổng quan

Các thuật giải tối ưu bầy đàn bao gồm một số thuật giải như: Thuật giải di truyền (Genetic Algorithm - GA), thuật giải bầy kiến (Ant Colony Algorithm - ACO), thuật giải tối ưu hóa bầy đàn (Particle Swarm Optimization - PSO). Một đặc điểm chung của các thuật giải tối ưu dựa trên bầy đàn là cố gắng tìm ra các thay đổi của lời giải đang xét để tối ưu cho lần lặp tiếp theo. Một số phương pháp tìm kiếm sử dụng điều kiện “tham lam” (greedy criterion) để quyết định là có nên giữ lại lời giải đang xét hay chấp nhận một lời giải mới. Bài toán người du lịch (TSP) là một bài toán cổ điển thuộc lớp NP được nghiên cứu sâu trong lĩnh vực tối ưu tổ hợp. Thuật toán bầy ong là một phương pháp theo hướng tiếp cận như trên. Bên cạnh các thuật giải dựa trên bầy

(*) ThS. Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh, hacnp@hufi.edu.vn

(**) ThS. Trường Đại học Văn Lang, minh.ntn@vlu.edu.vn

(***) ThS. Trường Đại học Văn Lang, dong.tv@vlu.edu.vn, Mã số: TCKH27-12-2021

đàn kể trên, một thuật giải tối ưu do tác giả Duc Truong Pham và cộng sự [5], [6] đề xuất là Thuật giải tối ưu BA (Bees Algorithm), thuật giải này mô phỏng dựa trên hành vi tìm kiếm thức ăn theo đàn của bầy ong.

2.2. Giới thiệu thuật toán tối ưu BA (Bees Algorithm)

Thuật toán tối ưu BA là thuật giải tối ưu dựa trên tối ưu hóa tổ hợp, được ứng dụng trong nhiều bài toán thuộc các lĩnh vực khác nhau: Lập lịch làm việc cho máy sản xuất [1], bài toán người du lịch, giải bài toán tối ưu hóa đa mục tiêu [2], huấn luyện mạng nơron cho nhận dạng mẫu [3]... Thuật toán tối ưu BA tái hiện lại các hành vi tìm kiếm thức ăn của bầy ong trong tự nhiên được xử lý ở mỗi bước lặp, kết quả sau khi xử lý cũng là một quần thể. Nếu bài toán chỉ có duy nhất một lời giải tối ưu, các cá thể trong quần thể của thuật toán tối ưu dựa trên bầy đàn có thể hội tụ quanh lời giải tối ưu. Nếu một bài toán có nhiều lời giải tối ưu, thuật toán tối ưu bầy đàn có thể được sử dụng để lấy được ở quần thể cuối cùng [4]. Thuật toán tối ưu BA sử dụng kết hợp tìm kiếm lân cận (Neighbourhood Search) và tìm kiếm ngẫu nhiên (Random Search).

Bees Algorithm = Neighbourhood Search + Random Search

2.3. Hành vi tìm kiếm thức ăn của bầy ong trong tự nhiên

Đầu tiên, ong do thám được cử đi tìm các bụi hoa. Toán ong do thám bay ngẫu nhiên trong không gian từ bụi hoa này đến bụi hoa khác. Khi trở về tổ, ong do thám nào tìm thấy bụi hoa nằm trên ngưỡng chất lượng nhất định (được đo bằng cách kết hợp các yếu tố như: hàm lượng đường, khoảng cách, hướng...), tỷ lệ mật hoa hoặc phấn hoa sẽ được đi đến “sàn nhảy” để thực hiện một điệu nhảy lúc lắc. Ong do thám thông qua điệu nhảy lúc lắc, truyền các thông tin về bụi hoa cho các ong khác. Các thông tin bao gồm: hướng được tìm thấy là góc giữa mặt trời và bụi hoa; khoảng cách từ tổ ong đến bụi hoa là thời gian của

điệu nhảy; chất lượng của bụi hoa là tần số của điệu nhảy. Các thông tin trên giúp bầy ong xác định được vị trí bụi hoa chính xác để thu thập thức ăn hiệu quả và nhanh chóng.



Hình 1. Cách xác định hướng, khoảng cách và chất lượng bụi hoa

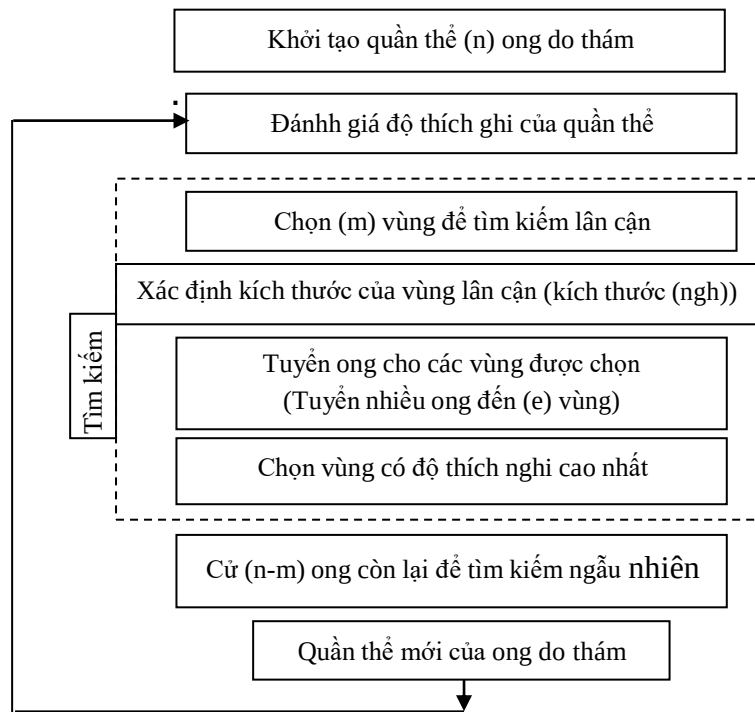
Khi thu hoạch tại bụi hoa, bầy ong sẽ giám sát và tính lượng thức ăn để quyết định trong “điệu nhảy lúc lắc” kế tiếp khi trở về tổ. Nếu các bụi hoa vẫn còn nhiều thức ăn, chúng sẽ thể hiện trong điệu nhảy và sẽ cử nhiều ong hơn đến bụi hoa này. Như vậy, từ điệu nhảy này sẽ quyết định có nên hay không nên cử ong tiếp tục đến đó [4], [5].

2.4. Mô hình hóa thuật toán tối ưu BA

Thuật toán tối ưu bầy ong lấy cảm hứng từ việc tìm kiếm thức ăn của bầy ong để tìm và cải thiện các giải pháp hiện tại bằng các giải pháp tốt nhất cho những vấn đề tối ưu hóa nhất định. Mỗi điểm trong không gian tìm kiếm (giải pháp tiềm năng) được coi là một nguồn thức ăn. Ong do thám ban đầu tìm thức ăn ngẫu nhiên (lời giải ngẫu nhiên được khởi tạo ban đầu) và dựa trên độ thích nghi các ong này báo cáo chất lượng của các vùng được thăm (đánh giá các lời giải). Các giải pháp được sắp xếp theo độ thích nghi và cử các ong đến những bụi hoa có độ thích nghi cao nhất, cử những ong còn lại tìm kiếm lân cận để tìm các lời giải khác (có thể sẽ tốt hơn). Thuật toán tối ưu BA chọn các giải pháp tốt nhất để tối ưu hóa hàm mục tiêu (tức chọn các giải pháp tốt nhất để làm giảm chi phí) [6], [7], [8].

2.5. Lưu đồ thuật toán tối ưu BA (Basic_Bees_Algorithm)

Từ những biểu diễn trên, chúng tôi có các bước mã giả cho thuật toán tối ưu BA như hình 3 [4].



Hình 2. Lưu đồ thuật toán tối ưu BA

1. Tạo ngẫu nhiên n vị trí (tương ứng với n scout bee)
2. Chọn m vị trí (bất kỳ) trong số n vị trí trên ($m \leq n$) để khảo sát
3. Tính $f(x)$ cho m vị trí này, chọn ra e vị trí có giá trị $f(x)$ tốt nhất
4. Ở mỗi vị trí $e_i \in e$, tính $f(x)$ của n vùng lân cận. Vùng lân cận một điểm A được xác định là 1 điểm ngẫu nhiên trong khoảng $x_A \pm ngh$
5. Ở mỗi vị trí $m_i \in (m - e)$, tính $f(x)$ của n vùng lân cận
6. Ở mỗi vị trí $n_i \in (n - m)$ còn lại, tạo ngẫu nhiên lại và đánh giá giá trị $f(x)$
7. Nếu điều kiện dừng chưa thỏa, quay lại 2

Hình 3. Các bước mã giả cho thuật toán tối ưu BA

2.6. Mô tả các bước của thuật toán

Bước 1: Thuật giải bắt đầu tạo ngẫu nhiên (n) vị trí - tương ứng (n) ong do thám trong không gian tìm kiếm;

Bước 2: Đánh giá độ thích nghi các vùng được thăm bởi các ong do thám khi quay về;

Bước 3: Xét điều kiện dừng chưa thỏa thì làm.

Bước 4: Chọn vùng để tìm kiếm lân cận. Các ong có độ tương thích cao nhất sẽ được chọn để tìm kiếm lân cận (neighbourhood search).

Bước 5, 6: Thuật giải tiến hành tìm kiếm lân cận của các vùng được chọn, phân công nhiều

ong hơn để tìm kiếm với (e) vùng tốt nhất. Các ong có thể được chọn trực tiếp dựa trên độ thích nghi kết hợp với các vùng mà chúng đã thăm. Giá trị độ thích nghi còn được sử dụng để xác định độ thích nghi xác suất các ong được chọn. Trong quá trình tìm kiếm lân cận của các vùng (e) tốt nhất đại diện cho những lời giải hứa hẹn hơn. Trong bước này, kích thước (ngh) được xác định sẽ được sử dụng để cập nhật số ong đến (m) vùng đã chọn ở bước trước đó. Điều này rất quan trọng vì có thể có những lời giải tốt hơn lời giải ban đầu trong các vùng lân

cận. Kết hợp với việc do thám, việc tuyển thêm ong khác nhau là thao tác chính của thuật giải bầy ong. Bước 6, trong mỗi khu vực chỉ có 1 ong với độ thích nghi cao nhất sẽ được chọn để tham gia vào quần thể tiếp theo. (Lưu ý: Trong tự nhiên không có giới hạn, giới hạn giới thiệu ở đây để làm giảm số vùng cần do thám);

Bước 7: Các ong còn lại trong quần thể được phân ngẫu nhiên do thám quanh không gian tìm kiếm, cho ra những lời giải tiềm năng mới. Đây chính là quá trình tìm kiếm ngẫu nhiên;

Quá trình từ bước 2 đến bước 7 lặp đi lặp lại cho đến khi thỏa điều kiện dừng. Tại cuối

mỗi bước lặp, quần thể ong sẽ có 2 phần đại diện cho quần thể mới là các cá thể tốt nhất cho mỗi vùng được chọn và những ong do thám được phân công để tiến hành tìm kiếm ngẫu nhiên. Thuật toán tối ưu BA bao gồm các tham số sau: (n): Số ong do thám (tương ứng với mỗi vùng); (m): Số vùng được chọn trong n vùng được thăm dò; (e): Số vùng tốt nhất trong m vùng; (nep): Số ong được cử đến (e) vùng tốt nhất; (nsp): Số ong được cử đến các vùng còn lại; (ngh): Kích thước của bụi hoa và lân cận của nó.

2.7. Các hàm số kiểm tra

Bảng 1. Giá trị chuẩn của các hàm số thực đạt min

Hàm f1: Rosenbrock's valley	
Công thức	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$
Miền xác định x_i	$-2.048 \leq x_i \leq 2.048, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 1 (i = 1, \dots, n.)$
Hàm f2: Rastrigin's function	
Công thức	$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$
Ngưỡng giá trị x_i	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 0, i = 1, \dots, n.$
Hàm f3: Schwefel's function	
Công thức	$f(x) = \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$
Ngưỡng giá trị x_i	$-500 \leq x_i \leq 500, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = -418.9829 \cdot n$
Argmin	$x_i = 420.9687, i = 1, \dots, n.$
Hàm f4: Sum of different power functions	
Công thức	$f(x) = \sum_{i=1}^n x_i ^{i+1}$
Ngưỡng giá trị x_i	$-1 \leq x_i \leq 1, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 0, i = 1, \dots, n.$

<i>Hàm f5: Griewangk's function</i>	
Công thức	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Ngưỡng giá trị x_i	$-600 \leq x_i \leq 600, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 0, i = 1, \dots, n$
<i>Hàm f6 :Ackley's function</i>	
Công thức	$f(x) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1)$
Ngưỡng giá trị x_i	$-32.768 \leq x_i \leq 32.768, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 0, i = 1, \dots, n.$
<i>Hàm f7 :Axis parallel hyper-ellipsoid function</i>	
Công thức	$f(x) = \sum_{i=1}^n (i \cdot x_i^2)$
Ngưỡng giá trị x_i	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, n.$
Giá trị tối ưu đạt được	Global minimum $f(x) = 0$
Argmin	$x_i = 0, i = 1, \dots, n.$

2.7.1. Mô tả thí nghiệm

Các hàm số được sử dụng để thực nghiệm thuật toán đều là những hàm số chuẩn (Benchmark Function) thường được sử dụng để đánh giá hiệu quả của các thuật toán tối ưu. Tất cả các hàm số này được tham khảo từ tài liệu [9]; Bảy hàm số trên được sử dụng trong chương trình thực nghiệm để minh họa cho các thuật toán: Trong phần này, chúng tôi trình bày chi tiết về dạng hàm của các hàm số này cũng như là hình minh họa các hàm đó trong không gian hai chiều sử dụng phần mềm Maple 15 (64 bit) và hàm Jfree char như là công cụ để vẽ hàm. Để cài đặt các thuật toán tối ưu BA, chúng tôi sử dụng ngôn ngữ Java, công cụ lập trình Net Bean IDE 8.0.2 các tham số của các thuật toán đều được chọn đồng nhất. Cụ thể như sau:

Scout Bees: Nhập số ong do thám; **Select Sites:** Số vùng do thám cho là tốt có tiềm năng; **Best Sites:** Số vùng tốt nhất trong Select Sites; **nep:** Số ong tới vùng Best Sites; **nsp:** Cử ong tới vùng tiềm năng còn lại; **ngb:** Kích thước bụi hoa với vùng lân cận; **Iterations (vòng lặp):** Chọn số lần kiểm tra; **Dimensions (khoảng cách giữa các con ong):** Số biến (n chiều của hàm tối ưu);

Bounds: Bán kính trục tọa độ xét; **Function:** Chọn một hàm tối ưu trong 7 hàm; **Expected:** Thể hiện giá trị và điểm nhận từ khoảng cách giữa các con ong; **Obtained:** Thể hiện giá trị tối ưu và giá trị trung bình cho 30 lần chạy và thời gian trung bình; **Solution of:** Đồ thị thể hiện thời gian chạy; **Process chart:** Đồ thị thể hiện fbest; Số lần chạy cho mỗi cấu hình là 30 lần; Mỗi hàm đều có thể chạy số chiều lần lượt là 5, 10, 15, và 20...

2.7.2. Kết quả thực nghiệm

Bảng 2, trình bày giá trị tối ưu trung bình tìm được bởi các thuật toán với 7 bài toán tối ưu hàm ở trên. Số chiều thử nghiệm của các hàm số từ 5 đến 20. Kết quả Bảng 2 cho thấy với số chiều càng lớn, thuật toán BA cho kết quả tốt, khi số chiều nhỏ, giá trị tối thiểu của thuật toán càng nhỏ. Kết quả thay đổi số chiều khi chạy 30 lần với số chiều là 5, 10, 15, 20 ta tính được giá trị trung bình tối ưu của từng thuật toán với số số ong ban đầu là 200, nep=12, nsp=10, ngb=1, số vòng lặp là 200 lần (bảng 2).

Kết quả trung bình thời gian của 30 lần chạy của thuật toán với số chiều là 5 và số ong là 200 (bảng 3).

Bảng 2. Trình bày giá trị thời gian trung bình trong 30 lần chạy

Hàm	Thuật toán	Số chiều			
		5	10	15	20
f1	BA	2.99677412	38.46743877	116.97518926	266.11783890
f2		5.22712866	38.43441019	81.36179489	141.51599852
f3		2010.23830296	3448.17716646	4776.53979358	-5860.45789304
f4		0.00052620	0.00800053	0.02030942	0.04087575
f5		2.25121020	23.30162078	62.05464684	126.19653017
f6		0.59918722	2.19246479	10.13536604	17.18948995
f7		0.07952524	1.88550574	8.47736657	22.93607239

Bảng 3. Giá trị thời gian trung bình/số chiều

Hàm	Thuật toán	Kết quả trung bình thời gian/Số chiều			
		5	10	15	20
f1	BA	682.80000000	1484.20000000	2173.56666667	2863.30000000
f2		676.10000000	1333.86666667	1978.30000000	2807.36666667
f3		623.00000000	1250.83333333	1845.23333333	2540.26666667
f4		616.73333333	1231.60000000	1821.83333333	2469.56666667
f5		671.46666667	1333.26666667	2102.73333333	2682.26666667
f6		689.23333333	1355.66666667	2002.03333333	2695.16666667
f7		621.23333333	1236.53333333	1832.30000000	2492.36666667

Như vậy, khi thay đổi số chiều trong thuật toán, kết quả tối ưu chưa chính xác vì độ sai số của hàm càng cao.

2.8. Ảnh hưởng của các tham số đến hiệu quả của thuật toán

Đánh giá ảnh hưởng của các tham số đầu vào là số ong tham gia làm ảnh hưởng đến hiệu

quả của các thuật toán. Thực nghiệm kết quả khi ta thay đổi số ong tham gia lần lượt thay đổi là 15, 50, 200, 500, với khoảng cách giữa các con ong là 5, số vòng lặp là 200, số lần kiểm tra là 30 lần.

Bảng 4. Giá trị khi thay đổi số ong tham gia

Hàm	Thuật toán	Số ong tham gia			
		15	50	200	500
f1	BA	3.21833282	3.16051406	2.99677412	2.73114902
f2		7.12094961	5.51141923	5.22712866	5.24087061
f3		1613.33795420	1955.49883181	2010.23830296	-2046.95107835
f4		0.00105527	0.00109006	0.00052620	0.00043791
f5		53.43781695	4.62154663	2.25121020	1.64170319
f6		10.99349184	0.64229946	0.59918722	0.57836461
f7		0.08082411	0.06848750	0.07952524	0.08545221

Bảng thời gian thực nghiệm kết quả, khi thay đổi số ong tham gia lần lượt là 15, 50, 200, 500, với khoảng cách giữa các con ong là 5, số vòng lặp là 200, số lần kiểm tra là 30 lần (Bảng 5). Như vậy, khi thực nghiệm thay đổi số ong tham gia trong thuật toán, nhìn chung nếu thay đổi số

ong có ảnh hưởng lớn cho ra kết quả tối ưu, đặc biệt trong các hàm thực nghiệm có hàm f3 (*Schwefel's function*) có giá trị rất cao là do nguyên thủy hàm có độ phức tạp (nếu độ phức tạp hàm càng cao, sai số càng lớn).

Bảng 5. Bảng thời gian thực nghiệm kết quả khi ta thay đổi số ong tham gia

Hàm	Thuật toán	Thời gian/Số ong tham gia/khoảng cách giữa các con ong là 5			
		15	50	200	500
f1	BA	333.53333333	400.60000000	682.80000000	1277.20000000
f2		326.90000000	391.73333333	676.10000000	1254.53333333
f3		304.50000000	364.23333333	623.00000000	1168.43333333
f4		302.66666667	363.86666667	616.73333333	1284.80000000
f5		325.36666667	394.36666667	671.46666667	1254.10000000
f6		336.80000000	404.30000000	689.23333333	1296.80000000
f7		344.33333333	363.50000000	621.23333333	1162.70000000

Như vậy, khi thực nghiệm thay đổi số ong tham gia trong thuật toán, nhìn chung nếu thay đổi số ong có ảnh hưởng lớn cho ra kết quả tối ưu, đặc biệt trong các hàm thực nghiệm có hàm f3 (*Schwefel's function*) có giá trị rất cao là do nguyên thủy hàm có độ phức tạp (nếu độ phức tạp hàm càng cao, sai số càng lớn).

2.9. Thực hiện so sánh khi thay đổi số vòng lặp

Thể hiện kết quả của các giá trị tối ưu khi thay đổi số vòng lặp, tức số lần tìm kiếm giá trị tối ưu. Thực nghiệm kết quả khi ta thay đổi số vòng lặp thay đổi là 10, 50, 100, 200, với khoảng cách giữa các con ong là 5, số ong tham gia cho mỗi thuật toán là 200, số lần kiểm tra là 30 lần (Bảng 6).

Bảng 6. Kết quả của các giá trị tối ưu khi thay đổi số vòng lặp

Hàm	Thuật toán	Số vòng lặp			
		10	50	100	200
f1	BA	7.98324140	4.67756891	3.89824140	2.99677412
f2		14.46507858	7.22689541	6.75261599	5.22712866
f3		1450.72960393	1802.90515032	1985.06181821	-2010.23830296
f4		0.00527734	0.00213983	0.00128173	0.00052620
f5		7.95126188	4.57024349	2.97885588	2.25121020
f6		8.62115433	1.18653679	0.79658143	0.59918722
f7		0.31664979	0.12134767	0.09322070	0.07952524

Bảng 7. Thời gian thực nghiệm kết quả khi ta thay đổi số vòng lặp

Hàm	Thuật toán	Thời gian/Số vòng lặp			
		10	50	100	200
f1	BA	35.90000000	180.66666667	348.13333333	682.80000000
f2		34.20000000	186.80000000	341.26666667	676.10000000
f3		31.90000000	176.93333333	320.96666667	623.00000000
f4		33.00000000	157.70000000	315.66666667	616.73333333
f5		34.26666667	177.43333333	339.50000000	671.46666667
f6		35.80000000	178.76666667	350.43333333	689.23333333
f7		33.73333333	162.86666667	319.40000000	621.23333333

Thời gian thực nghiệm cho kết quả khi thay đổi số vòng lặp thay đổi là 10, 50, 100, 200, với khoảng cách giữa các con ong là 5, số ong tham gia cho mỗi thuật toán là 200, số lần kiểm tra là 30 lần (Bảng 7).

Như vậy, với sự thay đổi về số vòng lặp, kết quả thể hiện như bảng thống kê, tuy thời gian xử lý lâu, nếu thay đổi số vòng lặp, tức số lần tìm kiếm thức ăn của mỗi con ong sẽ kỹ hơn.

3. KẾT LUẬN

Bài viết đã tìm hiểu tổng quan về tối ưu hóa nói chung và tối ưu tổ hợp trong đó có các kỹ thuật tối ưu hàm số thực. Tìm hiểu lý thuyết cũng như mô hình hóa, phân tích và đánh giá những ưu nhược điểm của thuật toán tối ưu BA cho vấn đề tối ưu tổ hợp. Mở rộng và cải thiện thuật toán tối ưu BA để giải quyết bài toán

phân cụm với các kiểu dữ liệu khác nhau. Nghiên cứu, cải thiện khả năng tìm kiếm lân cận trong thuật toán tối ưu BA. Xây dựng các mô hình thuật giải kết hợp giữa thuật toán tối ưu BA và các thuật giải khác như PSO, ACO, K- mean, Tabu Search... Xây dựng các ứng dụng thực tế cho vấn đề phân cụm dữ liệu.

TÀI LIỆU THAM KHẢO

- [1] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M (2005), *The Bees Algorithm*, Technical Note, Manufacturing Engineering Centre, Cardiff University, UK.
- [2] D. T. Pham, E. Koç, J. Y. Lee, and J. Phrueksanant (2007), *Using the Bees Algorithm to schedule jobs for a machine*. Proc Eighth International Conference on Laser Metrology, CMM and Machine Tool Performance, LAM DAMAP, Euspen, UK, Cardiff.
- [3] Pham D. T., Zaidi Muhamad, Massudi Mahmuddin, Afshin ghanbarzadeh, Ebubekir Koc, Sameh Otri. *Using the bees algorithm to optimise a support vector machine for wood defect classification, The bees algorithm: Modelling foraging*, IPROMS 2007 Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.
- [4] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M (2005), *The Bees Algorithm. Technical Note, Manufacturing Engineering Centre*. Cardiff University, UK.
- [5] Pham, D.T., Soroka, A.J., Koc, E., Ghanbarzadeh, A., Otri, S., Packianather, M. (2006), *Optimising neural networks for identification of wood defects using the bees algorithm*, In: Proceedings of the 4th International IEEE Conference on Industrial Informatics, INDIN.
- [6] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M. (2006), *The bees algorithm - a novel tool for complex optimisation problems*. In: Proceedings of IPROMS 2006 Conference.
- [7] Pham, D.T., Koc, E., Ghanbarzadeh, A., Otri, S. (2006), *Optimisation of the weights of multi-layered perceptrons using the bees algorithm*. In: IMS 2006 Intelligent Manufacturing Systems Conference.
- [8] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M. (2006), *The bees algorithm - a novel tool for complex optimisation problems*. In: Proceedings of IPROMS 2006 Conference.
- [9] Pham D. T., Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M (2005), *The Bees Algorithm*. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK.
- [10] Duc Truong Pham, Ashraf Afify, Ebubekir Koc, *Manufacturing cell formation using the Bees Algorithm*". IPROMS 2007 Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.
- [11] Pham, D.T., Karaboga, D. (2000), *Intelligent Optimisation Techniques*. Springer, London.
- [12] Wedde, H.F., Farooq, M., Zhang, Y. (2004), *BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior, ant colony, optimization and swarm intelligence*. In: Proceedings of the 4th International Workshop, Brussels, Belgium.

Ngày nhận bài: 05-4-2021. Ngày biên tập xong: 05-5-2021. Duyệt đăng: 20-5-2021