

GIẢI THUẬT CHAOTIC VORTEX SEARCH CHO BÀI TOÁN TỐI ƯU TOÀN CỤC

TRƯỜNG KHẮC TÙNG¹, ĐỖ HÀ PHƯƠNG¹, DƯƠNG ĐỨC HƯNG²

¹ Khoa Công Nghệ Thông Tin, Trường Đại Học Công Nghiệp Tp Hồ Chí Minh, Việt Nam

² Đại học Huế, 03 Lê Lợi, Huế, Việt Nam
tungtk@iuh.edu.vn

Abstract. Trong bài báo này, dựa trên nghiên cứu lý thuyết về giải thuật tìm kiếm tối ưu Vortex Search (VS), lý thuyết và ứng dụng lý thuyết Chaos vào họ giải thuật MetaHeuristics. Chúng tôi đề xuất cải tiến giải thuật VS bằng cách lai quy luật phát sinh tập ứng viên giải thuật VS với hàm Chaotic Bernoulli Map. Kết quả kiểm chứng trên tập 20 hàm Benchmark cho thấy giải thuật mới có kết quả tốt hơn so với nguyên bản trên các tiêu chí đánh giá.

Keywords. Global optimization; Artificial intelligence; Chaotic number; Hybrid algorithm;

CHAOTIC VORTEX SEARCH ALGORITHM FOR GLOBAL NUMERICAL OPTIMIZATION

Abstract. In this paper, based on theoretical studies of the optimal search algorithm Vortex Search (VS), the theory and application of Chaos theory to the Metaheuristics. We propose to improve the VS algorithm by hybridization of the rule for generating the candidate solutions algorithm of VS with chaotic function by using the Bernoulli map. Simulation results on a set of 20 Benchmark validation algorithms show that the new algorithm has better results than the old algorithm on the evaluation criterias.

Keywords. Global optimization; Artificial intelligence; Chaotic number; Hybrid algorithm;

1 MỞ ĐẦU

Trong những năm gần đây, đã có rất nhiều nghiên cứu lý thuyết, ứng dụng và cải tiến các giải thuật tìm kiếm xấp xỉ tối ưu họ Metaheuristics[1][2][3][4][5]. Hai vấn đề chính của giải thuật luôn gặp phải[2] là chi phí thực thi và dễ rơi vào bẫy cục bộ địa phương làm ảnh hưởng trực tiếp đến chất lượng lời giải. Đối với việc giải quyết vấn đề cục bộ địa phương thì ngoài hướng tiếp cận nghiên cứu tìm cảm hứng bầy đàn[2][5][6] trong tự nhiên, hướng đa bầy đàn Multi-Swarm[7] còn có một cách tiếp cận cải tiến các giải thuật này là ứng dụng lý thuyết hỗn loạn[1][3][4] cũng đang rất được quan tâm nghiên cứu. Trong số các hàm chaos thì hàm Chaotic Bernoulli Map được sử dụng khá phổ biến trong việc tạo ra chuỗi số thay thế bộ sinh số ngẫu nhiên. Trong bài báo này tập trung vào kết hợp hàm Chaotic Bernoulli Map và giải thuật VS và chứng minh tính hiệu quả của giải thuật mới thông qua 20 hàm Benchmark. Trong mục này, chúng tôi trình bày tóm lược về các nội dung: Bài toán tối ưu toàn cục hàm đại số, tóm lược về Metaheuristics, giải thuật Vortex Search và Lý thuyết hỗn loạn.

1.1 Bài toán tìm kiếm giải pháp tối ưu toàn cục

Cho hàm số $f: D \rightarrow R, D \subset R^n$; Tìm $x \in D$ để $f(x)$ đạt cực đại - Max hay cực tiểu - Min.

trong đó, R^n : Không gian số thực n chiều, D : miền ràng buộc hay chính là không gian tìm kiếm lời giải của bài toán.

Vectơ $x = (x_1, x_2, \dots, x_n)^T \in D \subset R^n$ được gọi là một giải pháp - Solution (một lời giải của bài toán tìm kiếm giải pháp tối ưu).

Giá trị $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in R^n$ được gọi là giải pháp tối ưu toàn cục nếu $x^* \in D$ và $f(x^*) \geq f(x), \forall x \in D$ với bài toán cực đại hóa hay $f(x^*) \leq f(x), \forall x \in D$ với bài toán cực tiểu hóa.

Vectơ $x' \in R^n$ được gọi là giải pháp tối ưu địa phương nếu $x' \in D$ và tồn tại một lân cận N_ϵ đủ nhỏ của điểm x' sao cho $f(x') \geq f(x), \forall x \in N_\epsilon \cap D$ với bài toán cực đại hóa hay $f(x') \leq f(x), \forall x \in N_\epsilon \cap D$ với bài toán cực tiểu hóa.

Dễ dàng nhận thấy điểm cực trị toàn cục cũng là một cực trị địa phương nhưng điều ngược lại không

hẳn đúng. Hàm chỉ có một cực trị duy nhất thuộc nhóm U: Unimodal, hàm có nhiều cực trị thuộc nhóm M: Multimodal. Nhóm Multimodal khiến bài toán tìm kiếm lời giải tối ưu khó hơn rất nhiều khi gặp vấn đề bẫy cục bộ địa phương.

Hàm phân tách được (Separate) có thể áp dụng tính toán tối ưu bằng cách tập trung hướng tìm kiếm vào từng biến thành phần và thực hiện xoay vòng trên từng biến. Trong khi đó, hàm không phân tách được (Non-Separate) phức tạp hơn rất nhiều khi hướng tìm kiếm phụ thuộc vào 2 hay nhiều biến thành phần.

Bộ 20 hàm số kiểm chuẩn (Benchmark) đầu vào kiểm chứng kết quả bài báo được phân bổ trên các nhóm hàm: U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable.

1.2 Tóm lược về Metaheuristics

Trở lại bài toán tối ưu toàn cục hàm đại số, có hai vấn đề chính của việc tìm kiếm giải pháp tối ưu toàn cục hàm đại số đó là vấn đề chi phí thực thi (thời gian, tài nguyên máy tính) và vấn đề bẫy cục bộ địa phương.

Đối với vấn đề chi phí thực thi, tiếp cận theo hướng xấp xỉ được nghiên cứu để giải quyết vấn đề. Thay vì tìm kiếm lời giải tối ưu chính xác thì sẽ đưa ra một giải pháp tốt nhất chấp nhận được đáp ứng được yêu cầu về thời gian cũng như năng lực máy tính[2][8].

Đối với vấn đề bẫy cục bộ địa phương, các tiếp cận như giải thuật tiến hóa, siêu trí thức Metaheuristic[2][8][9] được nghiên cứu để giải quyết vấn đề. Thay vì việc tìm kiếm các quy luật để phát hiện ra lời giải, các giải thuật Metaheuristics xác định không gian tìm kiếm lời giải, sau đó khởi tạo tập giải pháp ban đầu và thực hiện tiến hóa để làm tốt dần lên qua các thế hệ. Kết quả tốt nhất từ tập giải pháp sau cùng là lời giải của bài toán.

Họ giải thuật Metaheuristics tiến hành theo chiến lược thăm dò kết hợp với chiến lược khai thác. Thăm dò nhằm tránh bỏ sót các vùng tìm kiếm tiềm năng được thực hiện bằng cách đưa yếu tố ngẫu nhiên tham gia vào xây dựng quần thể cá thể tìm kiếm mới trong thế hệ kế tiếp. Khai thác nhằm tiếp cận nhanh nhất điểm cực trị vùng tìm kiếm có nghĩa là việc xây dựng quần thể tìm kiếm mới dựa trên những thông tin tốt nhất ở thế hệ hiện tại.

Vấn đề chính của Metaheuristic là cân bằng giữa việc thăm dò và khai thác, quá tập trung vào thăm dò khiến lời giải không chất lượng và ngược lại nếu quá tập trung vào khai thác lại khiến giải thuật dễ rơi vào bẫy cục bộ địa phương. Việc này được thực hiện bằng cách cố gắng cân bằng xác suất hướng vào việc thăm dò và hướng vào việc khai thác. Họ Metaheuristics gồm 2 lớp giải thuật: Single-Solution Based (Simulated Annealing SA[10], Random Search[11], Pattern Search[12], Vortex Search[5][13]...) và Population Based (Genetic Algorithm[8][9], Particle Swarm Optimization PSO[3][6][7], Social Spider Algorithm[14]...). Lớp giải thuật Single-Solution Based chỉ khởi tạo một cá thể tìm kiếm ban đầu, trong khi đó với Population Based là khởi tạo bầy đàn - một tập cá thể tìm kiếm.

1.3 Giải thuật Vortex Search

Giải thuật tìm kiếm xoáy Vortex Search (VS) được hai đồng tác giả Berat Doğan, Tamer Ölmez nghiên cứu và xuất bản[5][13] thuộc lớp giải thuật Single-Solution Based của họ giải thuật Metaheuristics. VS ngoài các điểm mạnh như thời gian thực thi nhanh và tốc độ hội tụ cao còn có các vấn đề về bẫy cục bộ địa phương.

Dưới đây là đoạn mã giả Psuedo-Code cho lớp giải thuật Single-Solution Based họ giải thuật Metaheuristics:

Input:

Khởi tạo cá thể giải pháp ban đầu s_0 ;

Bộ đếm thế hệ, ban đầu $t = 0$;

Repeat

Phát sinh tập cá thể ứng viên $C(s_t)$ dựa vào cá thể s_t ;

Chọn cá thể ứng viên tốt nhất $s_t: s_{t+1} = \text{Select}(C(s_t))$;

$s_{t+1} = \text{best}(s_t, s_{t+1})$;

Cập nhật thế hệ mới $t = t + 1$;

Until Thỏa điều kiện dừng;

Output: Giải pháp tối ưu s_t .

Single-Solution đại diện cho lời giải bài toán, các giải thuật họ Single-Solution Based Metaheuristics

gây dựng một cá thể Single-Solution ban đầu và làm tốt dần lên qua các thế hệ. Tại mỗi thế hệ, giải thuật phát sinh ra tập các cá thể giải pháp ứng viên theo một quy luật nào đó dựa vào Single-Solution và chọn ra ứng viên tốt nhất cho thế hệ đó, nếu ứng viên này tốt hơn lời giải bài toán hiện hành thì được bầu chọn thay thế làm lời giải bài toán.

Ý tưởng này áp dụng chung cho các giải thuật phát triển dựa trên Single-Solution Based Metaheuristics. Các giải thuật Single-Solution Based (Pattern Search (PS), Random Search (RS), Vortex Search (VS)...) phát triển Single-Solution dựa vào việc đưa ra quy luật phát sinh tập ứng viên như thế nào.

Giải thuật VS sử dụng quy luật phân phối chuẩn Gaussian-Distribution để phát sinh ra tập cá thể ứng viên. Phân bố Gauss bảo đảm rằng xác suất để phát sinh ra cá thể ứng viên tìm kiếm ở gần vị trí gần tâm là cao hơn xác suất phát sinh cá thể ở xa tâm tìm kiếm.

Dưới đây là đoạn mã giả Psuedo-Code giải thuật VS:

Inputs:

- Khởi tạo μ_0 tâm tìm kiếm ban đầu;
- Khởi tạo r_0 bán kính tìm kiếm ban đầu;
- Khởi tạo giá trị $f(S_{best}) = \inf$ hàm mục tiêu toàn cục;
- Khởi tạo bộ đếm lặp, ban đầu $t = 0$;

Repeat

Phát sinh tập cá thể ứng viên $C_t(s)$ sử dụng quy luật phân bố Gauss dựa vào tâm tìm kiếm μ_t ;

Điều chỉnh các cá thể $C_t(s)$ ngoài phạm vi vào vùng tìm kiếm;

Chọn giải pháp s^* tốt nhất từ tập ứng viên;

IF s^* tốt hơn S_{best}

THEN Cập nhật $S_{best} = s^*$;

ELSE Giữ S_{best} ;

END

Cập nhật tâm tìm kiếm $\mu_t = S_{best}$;

Giảm bán kính tìm kiếm r_{t+1} cho thế hệ $t+1$;

Tăng thế hệ tìm kiếm $t = t + 1$;

Until: Đạt số vòng lặp tối đa của giải thuật;

Output: Giá trị tối ưu $f(S_{best})$ giải thuật tìm được.

Dưới đây là các công thức được sử dụng cho giải thuật VS:

$$\mu_0 = \frac{\text{upperlimit} + \text{lowerlimit}}{2} \tag{1}$$

$$r_0 \approx \sigma_0 = \frac{\max(\text{upperlimit}) - \min(\text{lowerlimit})}{2} \tag{2}$$

$$s_k^i = \begin{cases} \text{rand}(\text{upperlimit}^i - \text{lowerlimit}^i) + \text{lowerlimit}^i; & s_k^i < \text{lowerlimit}^i \\ s_k^i; & \text{lowerlimit}^i \leq s_k^i \leq \text{upperlimit}^i \\ \text{rand}(\text{upperlimit}^i - \text{lowerlimit}^i) + \text{lowerlimit}^i; & s_k^i \geq \text{upperlimit}^i \end{cases} \tag{3}$$

Trong đó s_k^i là giá trị đặc trưng thứ i của cá thể thứ k .

$$r_t = \sigma_0 \cdot \left(\frac{1}{x}\right) \cdot \text{gammaincinv}(x, a_t) \tag{4}$$

$$a_t = a_0 - \frac{t}{\text{MaxInt}} \tag{5}$$

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\} \tag{6}$$

$$\Sigma = \sigma^2 \cdot [I]_{d \times d} \tag{7}$$

1.4 Lý thuyết Chaos

Lý thuyết Chaos được Henri Poincaré[15] phát hiện khi nghiên cứu về chuyển động của một tập đối tượng. Bài toán đặt ra là cho biết vị trí hiện tại và quy luật chuyển động, xác định vị trí của tập đối tượng tại một thời điểm trong tương lai. Gọi x là vị trí hiện tại của đối tượng, giả sử đối tượng chuyển động theo một hàm tác động còn gọi là hàm quy luật g , khi đó hàm quy luật chuyển động có dạng: $x_{t+1} = g(x_t)$; $t \in \mathbb{N}$. Trong đó $t + 1$ là thời điểm kế tiếp thời điểm t .

Khi nghiên cứu để xây dựng hệ thống dự báo từ những thông số đầu vào (vị trí hiện tại x_0 , các quy luật - tham số mô tả hàm f) có thể tiên đoán được kết quả trong tương lai (ứng dụng trong nghiên cứu dự báo), các nhà nghiên cứu đã phát hiện ra rằng ngoài đa số hàm quy luật có thể tiên đoán kết quả tương lai (hàm hội tụ về 1 giá trị hoặc vô cực) còn tồn tại những hàm rất nhạy cảm với đầu vào khiến cho kết quả rất hỗn độn trong một khoảng giá trị (không tiên đoán được).

Một hàm chuyển động Chaos có tính chất hỗn độn trong một khoảng xác định T là một ánh xạ liên tiếp từ T lên chính nó mà không tiên đoán được tương lai. Có nghĩa là không tiên đoán được sau một khoảng thời gian N thì chuyển động quay trở lại vị trí xuất phát $X = g^N(g(\dots g(X)))$. Các hàm số Chaotic map được xây dựng là hàm có ánh xạ hỗn độn trong khoảng $(0,1)$.

Lý thuyết Chaos được ứng dụng trong các hệ thống tính toán có yếu tố ngẫu nhiên tham gia. Ứng dụng lý thuyết Chaos với việc cải tiến quy luật hình thành yếu tố ngẫu nhiên là một hướng nghiên cứu trong những năm gần đây cho các giải thuật tìm kiếm tối ưu[1][4][5].

2 ĐỀ XUẤT GIẢI THUẬT CHAOTIC VORTEX SEARCH

Tìm hiểu về giải thuật VS cho thấy, việc nghiên cứu đề xuất giải thuật mới hay cải tiến các giải thuật họ Single-Solution Based Metaheuristic chính là việc phát hiện mới hay cải tiến các quy luật phát sinh tập ứng viên.

Những nghiên cứu ứng dụng lý thuyết Chaos vào cải tiến giải thuật Metaheuristic[1][3][4] cho thấy sự kết hợp Chaotic vào quy luật phát sinh làm cho giải thuật mới có kết quả tốt hơn dựa trên một số tiêu chí đánh giá. Từ việc nghiên cứu ứng dụng và cải tiến giải thuật VS, chúng tôi đề xuất giải thuật Chaotic Vortex Search (CHAOVS) dựa vào việc bổ sung quy luật Chaos vào hàm quy luật phát sinh tập ứng viên.

Hàm Chaotic được chọn là Bernoulli Map, được định nghĩa như sau:

$$s(n+1) = f_B(s(n))$$

Với điều kiện khởi tạo $s(0) \in [-1,1]$ và f_B là một ánh xạ chaotic $f_B: [-1,1] \rightarrow [-1,1]$ được định nghĩa:

$$f_B(s) = \begin{cases} \frac{2}{\alpha+1}s + \frac{1-\alpha}{\alpha+1}; & -1 \leq s < \alpha \\ \frac{2}{1-\alpha}s - \frac{1-\alpha}{1+\alpha}; & \alpha \leq s \leq 1 \end{cases}$$

Trong đó tham số $\alpha \in (-1,1)$. Giá trị $s(0)$ và α được sử dụng cho mô phỏng kết quả là $s(0) = 0.5$ và $\alpha = 0.8$.

2.1 Quy luật phát sinh tập ứng viên C_s giải thuật VS

Gọi μ_0 là tâm tìm kiếm ban đầu, μ_t là tâm tìm kiếm thế hệ thứ t .

Khi đó tâm $\mu_0 = [\mu_0^1, \mu_0^2, \dots, \mu_0^D]$, D là số chiều không gian tìm kiếm

Trong đó: $\mu_0^i = \frac{\text{lowerlimit}_i + \text{upperlimit}_i}{2}$,

$[\text{lowerlimit}_i, \text{upperlimit}_i]$ là ràng buộc cận dưới và cận trên không gian chiều thứ i . Các cận dưới và cận trên trên tất cả các chiều không gian với bộ hàm chuẩn Benchmark được thiết lập giá trị chung lowerlimit và upperlimit (xem bảng 4).

Tại mỗi thế hệ tìm kiếm, sử dụng hàm quy luật phân phối chuẩn phát sinh ma trận $C_{Psize,D}$ có $Psize$ dòng (kích thước quần thể) mỗi dòng chứa các giá trị số ngẫu nhiên theo quy luật phân phối Gauss (dựa theo công thức số (6), (7)) có giá trị trung bình Mean bằng 0, phương sai bằng 1:

$$C = \begin{bmatrix} \lambda_1^1 & \dots & \lambda_1^D \\ \vdots & \ddots & \vdots \\ \lambda_{Psize}^1 & \dots & \lambda_{Psize}^D \end{bmatrix}$$

Ma trận C được sử dụng như một mặt nạ tìm kiếm để xây dựng ma trận công cụ tìm kiếm C_t cho tâm tìm kiếm Mean bằng 0 độ lệch chuẩn r_t (là bán kính tìm kiếm thế hệ t):

$$C_t = \begin{bmatrix} r_t \lambda_1^1 & \cdots & r_t \lambda_1^D \\ \vdots & \ddots & \vdots \\ r_t \lambda_{Psize}^1 & \cdots & r_t \lambda_{Psize}^D \end{bmatrix}$$

Trong đó bán kính ban đầu r_0 được tính theo công thức (2):

$$r_0 = \frac{\max(upperlimit) - \min(lowerlimit)}{2}$$

Bán kính tìm kiếm được điều chỉnh giảm dần qua các thế hệ theo công thức điều chỉnh:

$$r_t = r_0 \cdot \left(\frac{1}{x}\right) \cdot \text{gammaincinv}(x, a_t); \text{ Với } a_0 = 1 \text{ và } a_t = 1 - \frac{t}{MaxInt};$$

$MaxInt$ là số lượng thế hệ giải thuật thực hiện tìm kiếm. Tham số x được hiểu như độ uốn của hệ số giảm bán kính. (Tham số x được sử dụng cho mô phỏng giải thuật là $x = 0.1$)

Ma trận công cụ tìm kiếm C_t được dịch chuyển tới vị trí tâm tìm kiếm thực sự (với tâm tìm kiếm μ_{t-1} sử dụng để phát sinh tập ứng viên thế hệ t):

$$C(s_t) = \begin{bmatrix} \mu_{t-1}^1 + r_t \lambda_1^1 & \cdots & \mu_{t-1}^1 + r_t \lambda_1^D \\ \vdots & \ddots & \vdots \\ \mu_{t-1}^{Psize} + r_t \lambda_{Psize}^1 & \cdots & \mu_{t-1}^D + r_t \lambda_{Psize}^D \end{bmatrix}$$

Trong đó $\mu_{t-1} = [\mu_{t-1}^1, \mu_{t-1}^2, \dots, \mu_{t-1}^D]$ là tâm tìm kiếm hiện tại sử dụng cho phát sinh tập ứng viên ở thế hệ mới $C(s_t)$.

Đặt $s_k^i = \mu_{t-1}^i + r_t \lambda_k^i$; các cá thể nếu vi phạm ràng buộc về chiều không gian thứ i (ràng buộc cận dưới và cận trên) sẽ được thực hiện điều chỉnh theo công thức (3).

Sau khi thực hiện điều chỉnh các cá thể vi phạm, ta có tập cá thể ứng viên thế hệ t giải thuật VS (gọi là $C^{VS}(s_t)$):

$$C^{VS}(s_t) = \begin{bmatrix} s_1^1 & \cdots & s_1^D \\ \vdots & \ddots & \vdots \\ s_{Psize}^1 & \cdots & s_{Psize}^D \end{bmatrix}$$

$C^{VS}(s_t)$ là một ma trận $Psize \times D$ trong đó $Psize$ là kích thước quần thể và D là số chiều không gian. Trong đó $s_i = (s_i^1, s_i^2, \dots, s_i^D)^T$ chính là cá thể ứng viên thứ i trong tập quần thể $Psize$ cá thể.

2.2 Quy luật phát sinh tập ứng viên C_s giải thuật CHAOVS

Sử dụng hàm Bernoulli map, phát sinh ma trận sử dụng để xây dựng quy luật phát sinh tập ứng viên mới:

$$\text{Bernoulli_map}(Psize, D) = \begin{bmatrix} c_1^1 & \cdots & c_1^D \\ \vdots & \ddots & \vdots \\ c_{Psize}^1 & \cdots & c_{Psize}^D \end{bmatrix} \text{ trong đó các } c_j^i \text{ là các giá trị số Chaotic phát sinh từ}$$

hàm Bernoulli map.

Chúng tôi thực hiện ghép với $C^{VS}(s_t)$ để tạo ra quy luật phát sinh tập ứng viên mới:

$$C^{CHAOVS}(s_t) = \begin{bmatrix} s_1^1 c_1^1 & \cdots & s_1^D c_1^D \\ \vdots & \ddots & \vdots \\ s_{Psize}^1 c_{Psize}^1 & \cdots & s_{Psize}^D c_{Psize}^D \end{bmatrix}$$

3 KẾT QUẢ THỰC NGHIỆM

3.1 Đầu vào giải thuật

Để kiểm tra kết quả thực nghiệm để so sánh đối chiếu giải thuật cũ, chúng tôi kiểm chứng trên đầu vào là 20 hàm số lấy từ bộ dữ liệu Benchmark Function[16]. Kiểm chứng thực thi giải thuật được thực hiện trên phần mềm MATLAB R2013a, cấu hình máy CPU: Intel Core I5 5300u 2.3GHz, bộ nhớ đệm 3MB Cache, RAM: 8GB - DDR3 1600MHz.

Bộ 20 hàm kiểm chứng được chọn ngẫu nhiên nhưng chứa các đặc điểm khác nhau trong bộ hàm chuẩn Benchmark: U-Unimodal (Hàm đơn cực trị), M-Multimodal (Hàm đa cực trị), S-Seperable (Hàm phân tách được), N-Non_Seperable (Hàm không phân tách được).

Bảng 1: 20 Hàm Benchmark kiểm chứng kết quả giải thuật dựa vào U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable

Func.	Name	C C: Characteristics	Mô tả
F_1	<i>Sphere</i>	US	$f(\mathbf{x}) = \sum_{i=1}^n (x_i)^2$
F_2	<i>Quartic</i>	US	$f(\mathbf{x}) = \sum_{i=1}^n (ix_i)^4 + \text{random}[0,1)$
F_3	<i>Easom</i>	UN	$f(\mathbf{x}) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
F_4	<i>Colville</i>	UN	$f(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$
F_5	<i>Trid6</i>	UN	$f(\mathbf{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$
F_6	<i>Zakharov</i>	UN	$f(\mathbf{x}) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$
F_7	<i>Schwefel 1.2</i>	UN	$f(\mathbf{x}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
F_8	<i>Foxholes</i>	MS	$f(\mathbf{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{\sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$
F_9	<i>Rastrigin</i>	MS	$f(\mathbf{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
F_{10}	<i>Michalewicz2</i>	MS	$f(\mathbf{x}) = -\sum_{i=1}^n \sin(x_i) (\sin(ix_i^2 / \pi))^{2m}$ $m = 10$
F_{11}	<i>Six Hump Camel Back</i>	MN	$f(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
F_{12}	<i>Shubert</i>	MN	$f(\mathbf{x}) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right)$
F_{13}	<i>Goldstein-Price</i>	MN	$f(\mathbf{x}) = \left[\frac{1 + (x_1 + x_2 + 1)^2}{(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)} \right] \cdot \left[\frac{30 + (2x_1 - 3x_2)^2}{(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)} \right]$
F_{14}	<i>Shekel5</i>	MN	$f(\mathbf{x}) = -\sum_{i=1}^5 \sum_{j=1}^4 \left[(x_j - a_{ij}) (x_j - a_{ij})^T + c_i \right]^{-1}$
F_{15}	<i>Powersum</i>	MN	$f(\mathbf{x}) = \sum_{k=1}^n \left[\left(\sum_{i=1}^n x_i^k \right) - b_k \right]^2$
F_{16}	<i>Hartman3</i>	MN	$f(\mathbf{x}) = -\sum_{i=1}^4 c_i \exp \left[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$
F_{17}	<i>Ackley</i>	MN	$f(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$
F_{18}	<i>Penalized</i>	MN	$f(\mathbf{x}) = \frac{\pi}{n} \left\{ + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$

			$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$
F_{19}	<i>Langerman5</i>	MN	$f(x) = - \sum_{i=1}^m c_i \left(\exp\left(-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2\right) \cos\left(\pi \sum_{j=1}^n (x_j - a_{ij})^2\right) \right)$
F_{20}	<i>Fletcherpowell10</i>	MN	$f(x) = \sum_{i=1}^n (A_i - B_i)^2$ $A_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j); B_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j)$

Để đánh giá khách quan giải thuật CHAOVS so với giải thuật VS nguyên bản, cả hai giải thuật sử dụng cùng các tham số đầu vào và cùng các tham số đầu vào các hàm sử dụng để kiểm chứng giải thuật (xem bảng 2 và bảng 3).

Bảng 2: Các tham số đầu vào giải thuật

Parameter	Variable name	Value
Số cá thể ứng viên	<i>Psize</i>	50
Số thế hệ	<i>maxIter</i>	30000

Bảng 3: Các tham số đầu vào hàm kiểm chứng giải thuật

Func.	Name	Lowerlimit	Upperlimit	Dim
F_1	<i>Sphere</i>	-100	100	30
F_2	<i>Quartic</i>	-1.28	1.28	30
F_3	<i>Easom</i>	-100	100	2
F_4	<i>Colville</i>	-10	10	4
F_5	<i>Trid6</i>	-36	36	6
F_6	<i>Zakharov</i>	-5	10	10
F_7	<i>Schwefel 1.2</i>	-100	100	30
F_8	<i>Foxholes</i>	-65.536	65.536	2
F_9	<i>Rastrigin</i>	-5.12	5.12	30
F_{10}	<i>Michalewicz2</i>	0	π	2
F_{11}	<i>Six Hump Camel Back</i>	-5	5	2
F_{12}	<i>Shubert</i>	-10	10	2
F_{13}	<i>Goldstein-Price</i>	-2	2	2
F_{14}	<i>Shekel5</i>	0	10	4
F_{15}	<i>Powersum</i>	0	4	4

F_{16}	<i>Hartman3</i>	0	1	3
F_{17}	<i>Ackley</i>	-32	32	30
F_{18}	<i>Penalized</i>	-50	50	30
F_{19}	<i>Langerman5</i>	0	10	5
F_{20}	<i>Fletcherpowell10</i>	$-\pi$	π	10

3.2 Kết quả mô phỏng

Số liệu ở bảng 5 là bảng tổng hợp kết quả đối chiếu thực thi giải thuật CHAOVS và VS:

Bảng 4: Bảng tổng hợp số liệu kiểm chứng kết quả giải thuật

<i>Run = 30.000</i>	<i>CHAOVS</i>			<i>VS</i>		
	<i>Mean</i>	<i>Std</i>	<i>Best</i>	<i>Mean</i>	<i>Std</i>	<i>Best</i>
F_1	4.81958E-33	8.6119E-33	4.64173E-35	5.65858E-25	2.06199E-24	9.32348E-28
F_2	0.005177838	0.002357641	0.001938333	0.008928602	0.004382526	0.002712584
F_3	-1	0	-1	-1	0	-1
F_4	3.69103E-10	7.13386E-10	1.97398E-13	2.39655E-09	6.59171E-09	3.74019E-14
F_5	-50	4.08389E-14	-50	-50	5.19466E-14	-50
F_6	1.88195E-43	9.80624E-43	9.67224E-51	6.35407E-36	1.3095E-35	3.23658E-44
F_7	2.61934E-09	2.75272E-09	9.28824E-11	8.23833E-05	7.18908E-05	5.80718E-06
F_8	0.998003838	1.16624E-16	0.998003838	0.998003838	1.23698E-16	0.998003838
F_9	74.29008899	18.17844465	39.79830178	24.67497051	10.01251071	1.989918114
F_{10}	-1.80130341	9.03362E-16	-1.80130341	-1.80130341	9.03362E-16	-1.80130341
F_{11}	-1.031628453	6.04589E-16	-1.031628453	-1.031628453	6.1849E-16	-1.031628453
F_{12}	-186.7309088	3.61826E-14	-186.7309088	-186.7309088	4.12208E-14	-186.7309088
F_{13}	3	1.59267E-15	3	3	2.05668E-15	3
F_{14}	-8.900922997	2.606047823	-10.15319968	-10.15319968	6.14463E-15	-10.15319968
F_{15}	0.000139549	0.000141825	3.40091E-07	0.000191344	0.000137016	4.1363E-07
F_{16}	-3.862782148	2.61169E-15	-3.862782148	-3.862782148	2.53907E-15	-3.862782148
F_{17}	0.10827178	0.450418703	2.22045E-14	9.69299E-14	5.54126E-14	5.06262E-14

F_{18}	5.481639022	9.703396842	3.9261E-31	2.444523822	3.201759124	9.25384E-28
F_{19}	-1.408932528	0.207328704	-1.499999223	-1.410832488	0.202791383	-1.499999223
F_{20}	13.49358084	36.43413327	4.21614E-08	23.47632067	63.71754452	7.80379E-05

Bảng số liệu so sánh các tiêu chí rút ra từ bảng tổng hợp kết quả, trong đó dấu +, =, - thể hiện CHAOVS có kết quả tốt hơn, ngang bằng và xấu hơn VS.

Bảng 5: Bảng tổng hợp so sánh trên các tiêu chí đánh giá

Run = 30.000	CHAOVS&VS		
	Mean	Std	Best
F_1	+	+	+
F_2	+	+	+
F_3	=	=	=
F_4	+	+	-
F_5	=	+	=
F_6	+	+	+
F_7	+	+	+
F_8	=	+	=
F_9	-	-	-
F_{10}	=	=	=
F_{11}	=	+	=
F_{12}	=	+	-
F_{13}	+	+	-
F_{14}	-	-	=
F_{15}	+	-	=
F_{16}	=	-	=
F_{17}	-	-	+
F_{18}	-	-	+
F_{19}	-	-	=
F_{20}	+	+	+

Kết quả bảng 5 cho thấy ngoài một số hàm tương đồng về kết quả so sánh trên các tiêu chí đánh giá ($F_1, F_2, F_3, F_6, F_7, F_9, F_{10}, F_{20}$), các hàm còn lại có sự khác nhau như hàm F_4 thắng trên tiêu chí giá trị bình quân Mean, tiêu chí độ lệch chuẩn nhưng lại thua ở tiêu chí về giá trị min tốt nhất Best.

Từ bảng tổng hợp so sánh tiêu chí giá trị bình quân, độ lệch chuẩn và giá trị tối ưu tốt nhất chúng tôi rút ra được bảng 6 tổng kết kết quả mô phỏng hai giải thuật.

Bảng 6: Bảng tổng hợp kết quả

<i>Run = 30.000</i>	CHAOVS&VS (+/=/-)		
	<i>Mean</i>	<i>Std</i>	<i>Best</i>
US	2/0/0	2/0/0	2/0/0
UN	3/2/0	4/1/0	2/2/1
MS	0/2/1	1/1/1	0/2/1
MN	3/3/4	4/2/6	3/5/2
Total	8/7/5	11/2/7	7/9/4

Kết quả cho thấy dựa vào tiêu chí giá trị bình quân Mean, giải thuật CHAOVS tốt hơn VS với 8 hàm so với 5 của VS. Tiêu chí giá trị tối ưu tốt nhất Best của CHAOVS khá tương đồng với VS.

4 KẾT LUẬN

Trong bài báo này, chúng tôi đã đưa ra các tóm lược lý thuyết về các vấn đề và giải quyết bài toán tối ưu toàn cục, lý thuyết hỗn động Chaos, giải thuật VS ứng dụng cho bài toán tối ưu toàn cục. Chúng tôi đã đưa ra một cách tiếp cận mới để cải tiến giải thuật VS bằng cách ứng dụng lý thuyết Chaos và đề xuất giải thuật mới Chaotic Vortex Search Algorithm và đã đạt được những kết quả nhất định. Những kết quả này là cơ sở để chúng tôi tiếp tục ứng dụng những hàm Chaos khác và kiểm chứng thực nghiệm để tiếp tục đi sâu vào hướng nghiên cứu đề xuất cải tiến giải thuật VS.

TÀI LIỆU THAM KHẢO

- [1] Sheikholeslami, R., and A. Kaveh. "A survey of chaos embedded meta-heuristic algorithms." *Int J Optim Civil Eng* 3.4 (2013): 617-33..
- [2] Rothlauf, Franz. Design of modern heuristics: principles and application. Springer Science & Business Media, 2011.
- [3] Li, Jun-wei, Yong-mei Cheng, and Ke-zhe Chen. "Chaotic particle swarm optimization algorithm based on adaptive inertia weight." *The 26th Chinese Control and Decision Conference (2014 CCDC)*. IEEE, 2014.
- [4] Bilal Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*. Vol 37, pp. 5682–5687, 2010.
- [5] Doğan B., Ölmez T. "A new Metaheuristic for numerical function optimization: Vortex Search algorithm," *Information Sciences*. Vol 293, pp. 125-145, ISSN 0020-0255, 1 February 2015.
- [6] Eberhart, Russell, and James Kennedy. "Particle swarm optimization." *Proceedings of the IEEE international conference on neural networks*. Vol. 4. 1995.
- [7] Zhao, Shi-Zheng, et al. "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization." *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008.
- [8] Xing, Bo, and Wen-Jing Gao. "Introduction to Computational Intelligence." *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Springer, Cham, 2014. 3-17.

- [9] Mitchell, Melanie, "An Introduction to Genetic Algorithms," Cambridge, MA: MIT Press, ISBN 9780585030944, 1996.
- [10] Granville, Vincent, Mirko Krivánek, and J-P. Rasson. "Simulated annealing: A proof of convergence." *IEEE transactions on pattern analysis and machine intelligence* 16.6 (1994): 652-656.
- [11] Rastrigin L.A.. "The convergence of the random search method in the extremal control of a many parameter system," *Automation and Remote Control*. Vol 24 (10), pp. 1337–1342, 1963.
- [12] Hooke R., Jeeves T. A. "Direct search" solution of numerical and statistical problems," *Journal of the Association for Computing Machinery (ACM)*. Vol 8 (2), pp. 212–229, 1961.
- [13] Doğan B., Ölmez T. "Modified Off-lattice AB Model for Protein Folding Problem Using the Vortex Search Algorithm," *International Journal of Machine Learning and Computing*. Vol. 5(4), pp. 329-333, 2015.
- [14] Dang, Binh Thanh, Minh Cong Vo, and Tung Khac Truong. "Social spider algorithm-based spectrum allocation optimization for cognitive radio networks." *International Journal of Applied Engineering Research* 12.13 (2017): 3879-3887.
- [15] Diacu, Florin, and Philip Holmes. *Celestial encounters: the origins of chaos and stability*. Vol. 22. Princeton University Press, 1999.
- [16] Liang, J. J., B. Y. Qu, and P. N. Suganthan. "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization." *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore* 635 (2013).

Ngày nhận bài: 29/03/2019

Ngày chấp nhận đăng: 03/01/2020