TRƯỜNG ĐẠI HỌC SÀI GÒN
TẠP CHÍ KHOA HỌC
ĐẠI HỌC SÀI GÒN
ISSN 1859 - 3208
Số 91, Tập 01/2024

SAIGON UNIVERSITY
SCIENTIFIC JOURNAL
OF SAIGON UNIVERSITY
No. 91, Vol 01/2024

Email: tcdhsg@sgu.edu.vn ; Website: http://sj.sgu.edu.vn/

TIẾP CẬN THUẬT TOÁN METAHEURISTIC GIẢI BÀI TOÁN HỆ PHƯƠNG TRÌNH PHI TUYẾN

Approach to metaheuristic algorithm solving nonlinear system of equations

Lê Tấn Long
Trường ĐH Sài Gòn

TÓM TẮT

Các bài toán tối ưu hóa phi tuyến luôn được các nhà khoa học quan tâm vì hầu hết các hệ thống trong thực tiễn đều có bản chất phi tuyến. Các hệ thống phi tuyến thường được xấp xỉ qua việc tuyến tính hóa, điều này khá chính xác trong một phạm vi giá trị đầu vào nhất định; nhưng khi hệ thống thay đổi nhỏ thì sẽ gây ra các hiệu ứng phức tạp. Trong bài báo này, chúng tôi đề xuất kết hợp thuật toán Grey wolf optimizer (GWO) và thuật toán Whale optimization (WOA) để nâng cao chất lượng lời giải bài toán hệ phương trình phi tuyến, chúng tôi đặt tên là thuật toán GWOWOA. Chúng tôi đã thực nghiệm thuật toán GWOWOA trên hai bài toán cụ thể F_1, F_2, kết quả thực nghiệm cho thấy rằng, chất lượng lời giải của thuật toán GWOWOA tốt hơn một số thuật toán hiện biết.

***Từ khóa**: metaheuristic, GWO, WOA, lai, hệ phương trình phi tuyến*

ABSTRACT

Nonlinear problems are always a concern for engineers and scientists because most natural systems are inherently nonlinear. Solving nonlinear systems of equations is often very complex and time-consuming. Nonlinear systems are often approximated by linearization, which is fairly accurate within a certain range of input values, but small changes in the system can cause complex effects. The author has chosen to combine the Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) to achieve better performance. GWO performs better with unimodal and multimodal benchmark functions. With two test problems, F_1 and F_2, the new GWOWOA algorithm shows more promising results compared to other algorithms in most cases. Regarding execution time, GWOWOA competes with GWO and WOA, never taking longer and sometimes being faster, as shown in problems F_1 and F_2. Thus, this algorithm is very promising for solving nonlinear systems of equations.

***Keywords**: metaheuristic, grey wolf, whale, crossover, system of nonlinear equations*

1. Giới thiệu bài toán tối ưu hóa phi tuyến và khảo sát một số hướng tiếp cận giải bài toán hệ phương trình phi tuyến

1.1. Giới thiệu bài toán hệ phương trình phi tuyến

Hệ phương trình phi tuyến không thể được viết dưới dạng tổ hợp tuyến tính của các biến [9].

Dạng của hệ phương trình phi tuyến:

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{cases}$$

Tác giả liên hệ: ltlong@sgu.edu.vn

Trong đó: f_1, f_2, …., f_n là các hàm phi tuyến phụ thuộc vào các biến x_1, x_2, …, x_n.

Ví dụ 1: Giải hệ phương trình phi tuyến

$$\begin{cases} f_1(x_1, x_2) = 2x_1 - e^{x_2} = 0 \\ f_2(x_1, x_2) = e^{-x_1} + 5x_2 = 0 \end{cases}$$

Ví dụ 2: Giải hệ phương trình phi tuyến

$$\begin{cases} f_1(x_1, x_2) = x^3 - 2xy + y^2 = 0 \\ f_2(x_1, x_2) = x^2 - 2x - y + 2 = 0 \end{cases}$$

1.2. Ứng dụng của bài toán tối ưu hóa phi tuyến

Hệ phương trình phi tuyến phát sinh trong nhiều lĩnh vực như kỹ thuật, cơ khí, y học, hóa học, robot, phân tích thị trường tài chính [1], [2].

1.3. Các hướng tiếp cận giải bài toán hệ phương trình phi tuyến

Hiện có các hướng tiếp cận giải giải bài toán hệ phương trình phi tuyến sau đây:

Thứ nhất, tiếp cận theo phương pháp số

đã được phát triển để giải hệ phi tuyến F(x) = 0, sử dụng đa thức Taylor, phân rã, nhiễu loạn đồng vị, công thức vuông góc và các kỹ thuật khác [3]. Phương pháp vùng tin cậy (Trust - Region Method) đảm bảo giải pháp bằng cách bắt đầu với hai giá trị ban đầu mà giải pháp tồn tại giữa hai giá trị đó. Tuy nhiên, sự hội tụ của các phương pháp này có thể chậm. Nhiều phương pháp số như phương pháp Newton đã được đề xuất, nhưng chúng có nhược điểm như yêu cầu thông tin dẫn xuất, điểm xuất phát ban đầu tốt và dễ bị mắc kẹt trong cực tiểu cục bộ [1].

Thứ hai, tiếp cận theo phương pháp biến bài toán phi tuyến thành bài toán tối ưu đa mục tiêu

Ngoài các phương pháp số, hệ phương trình phi tuyến đã được mô hình hóa thành các bài toán tối ưu hóa đa mục tiêu [3], [8]. Ưu điểm của việc mô hình hóa này là người dùng có thể tìm nhiều giải pháp thay thế thay vì chỉ một giải pháp duy nhất. Các bài toán tối ưu hóa được ứng dụng rộng rãi trong các ngành như sản xuất chế tạo, kỹ nghệ – cơ khí, kiểm soát điều khiển, xây dựng mô hình – chính sách, định tuyến mạng, tiếp thị, sắp xếp lịch trình, hàng không vũ trụ, tính toán – phát triển trí tuệ, vận trù học (operations research) [7].

Thứ ba, các thuật toán metaheuristic

Thuật toán Metaheuristic là các thuật toán siêu heuristic cho việc giải các bài toán khó. Bộ tối ưu hóa Metaheuristic bao gồm những chiến lược khác nhau trong việc khám phá không gian tìm kiếm, cân bằng giữa tính đa dạng và chuyên sâu. Một số hướng tiếp cận phổ biến sử dụng các thuật toán metaheuristic để giải bài toán phi tuyến là: Genetic Algorithm GA) [5], Particle Swarm Optimization(PSO) [2], Grey Wolf Optimizer(GWO) [6], Whale Optimization Algorithm(WOA) [4]. Hiện tại, thuật toán metaheuristic cho chất lượng giải tốt nhất, nhưng thời gian thực hiện chậm. Vấn đề đặt ra của chúng tôi trong bài báo này là mong muốn sử dụng thuật toán metaheuristic giải hệ phương trình cho chất lượng bài toán và thời gian thực thi với thời gian tốt nhất bằng cách sử dụng phương pháp kết hợp giữa thuật toán GWO và thuật toán WOA.

2. Thuật toán metaheuristic

Trong những năm gần đây, thuật toán metaheuristic thường được sử dụng để giải quyết những bài toán tối ưu và được đánh giá là cách tiếp cận tốt nhất hiện biết trong lĩnh vực khoa học máy tính trong việc giải quyết bài toán tối ưu.

2.1. Giới thiệu về thuật toán metaheuristic

Hầu hết các thuật toán metaheuristic đều lấy cảm hứng từ thiên nhiên, sử dụng các cơ chế và nguyên lý tự nhiên để thiết

kế các hệ thống tính toán nhân tạo nhằm giải quyết những vấn đề tính toán phức tạp và thường được dùng để tối ưu hóa toàn cục.

2.2. Các yếu tố chung của thuật toán metaheuristic bao gồm:

- Khởi tạo (initialization): Tạo ra một hoặc nhiều nghiệm ban đầu, thường là ngẫu nhiên.
- Đánh giá (evaluation): Sử dụng hàm đánh giá để đo lường chất lượng của các nghiệm.
- Lựa chọn (selection): Chọn các nghiệm tốt nhất hoặc tiềm năng từ tập hợp hiện tại.
- Thao tác (manipulation): Tạo nghiệm mới từ các nghiệm đã chọn.
- Cập nhật (update): Thay thế nghiệm hiện tại bằng nghiệm mới dựa trên chất lượng.
- Tăng cường (intensification): Tập trung cải thiện các giải pháp tốt đã biết.
- Đa dạng (diversification): Khám phá các vùng khác nhau trong không gian giải pháp.
- Kết thúc (termination): Xác định điều kiện dừng (số lần lặp, thời gian, cải thiện chất lượng).
- Khám phá và khai thác (exploration and exploitation): Cân bằng giữa tìm kiếm vùng mới và tối ưu hóa vùng đã biết.

2.3. Sơ đồ thuật toán metaheuristic cơ bản

Thuật toán metaheuristic đơn giản được thực hiện qua các bước sau:

1. Khởi tạo lời giải ban đầu được
định nghĩa các chiến lược tăng cường hoá.
2. Tính độ thích nghi cho mỗi lời giải.
3. Kiểm tra điều kiện kết thúc thuật toán.
4. Thực hiện các chiến lược tăng cường hoá.
5. Thực hiện các chiến lược đa dạng hoá.
6. Cập nhật lời giải hiện tại.
7. Chọn kết quả nếu thoả mãn điều kiện dừng thì trả với lời giải tốt nhất.

2.4. Giới thiệu thuật toán tối ưu hóa GWO và WOA

2.4.1. Thuật toán tối ưu hóa GWO

Việc thăm dò sau đó ra quyết định bao vây được biểu diễn như sau [9]:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \quad (3.1)$$

$$\vec{X}(t+1) = \vec{X}(t) - \vec{A}.\vec{D} \quad (3.2)$$

Với t là vòng lặp hiện thời, còn $\vec{A}$ và $\vec{C}$ là các vector hệ số. $\vec{A}$ là yếu tố do dự duy trì khoảng cách tìm kiếm với con mồi, $\vec{C}$ là hệ số hội tụ, sự trở ngại của hướng tiếp cận đến con mồi trong tự nhiên, cung cấp các giá trị ngẫu nhiên tại mọi thời điểm để tăng việc tìm kiếm trong các bước lặp [11].

Hai vector hệ số $\vec{A}$ và $\vec{C}$ sẽ được tính như sau:

$$\vec{A} = 2.\vec{a}.\overrightarrow{r_1} - \vec{a} \quad (3.2) \quad \vec{C} = 2.\overrightarrow{r_2}$$

$$\vec{a}=\left[1 - \frac{t}{t_{max}}\right]2$$

công thức cập nhật vị trí:

$$\overrightarrow{D_\alpha} = \left|\vec{C}.\overrightarrow{X_\alpha}(t) - \vec{X}(t)\right|$$
$$\overrightarrow{D_\beta} = \left|\vec{C}.\overrightarrow{X_\beta}(t) - \vec{X}(t)\right|$$
$$\overrightarrow{D_\delta} = \left|\vec{C}.\overrightarrow{X_\delta}(t) - \vec{X}(t)\right|$$
$$\overrightarrow{X_1} = \overrightarrow{X_\alpha}(t) - \vec{A}\overrightarrow{D_\alpha}$$
$$\overrightarrow{X_2} = \overrightarrow{X_\beta}(t) - \vec{A})\overrightarrow{D_\beta}$$
$$\overrightarrow{X_1} = \overrightarrow{X_\delta}(t) - \vec{A})\overrightarrow{D_\delta}$$
$$\vec{X}_{(T+1)} = \frac{\vec{X}1 + \vec{X}2 + \vec{X}3}{3} \quad (3.3)$$

Với $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$, $\overrightarrow{D_\delta}$là khoảng cách giữa cá thể tìm kiếm đối với sói alpha, sói beta, sói delta. $\overrightarrow{X_1}$, $\overrightarrow{X_2}$, $\overrightarrow{X_3}$ là vị trí di chuyển của con sói về phương hướng của sói alpha, sói beta, sói delta; Và $\vec{X}_{(t+1)}$ là vị trị mới của con sói sau khi cập nhật.

GWO Algorithm

1. Khởi tạo một quần thể cá thể tìm kiếm với các vị trí X_i (i=1,2,...,n)

2. Khởi tạo giá trị của các tham số $\vec{a}$, $\vec{A}$ và $\vec{C}$

3. Ước tính giá trị thích nghi của mỗi cá thể tìm kiếm

4. X_α = cá thể tìm kiếm tốt nhất

5. X_β = cá thể tìm kiếm tốt thứ hai

6. X_δ = cá thể tìm kiếm tốt thứ ba

7. Cho t = 0

8. while (t < t_{max})

9. for cá thể thứ i : trong Xs

10. Tính các giá trị $\vec{A}$ và $\vec{C}$

11. Cập nhật vị trí của X_i

12. End for

13. Ước tính giá trị thích nghi các mọi cá thể tìm kiếm

14. Cập nhật giá trị cho X_α, X_β, X_δ

15. Cập nhật tham số $\vec{a}$

16. t = t + 1;

17. return X_α

2.4.2. Thuật toán tối ưu hóa (WOA)

- Cơ chế săn mồi "lưới bong bóng" của WOA

Vì không thể biết được vị trí của con mồi lúc ban đầu nên WOA giả định rằng vị trí của cá thể có kết quả tốt nhất hiện tại là mục tiêu hoặc gần tối ưu.Hành vi này được thể hiện bởi những điều sau đây phương trình [10]:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)\right| \quad (3.4)$$

$$\vec{X}(t+1) = \overrightarrow{X*}(t) - \vec{A}.\vec{D}$$

Trong đó t là vòng lặp hiện thời.

$\vec{X}$* là vectơ vị trí của cá thể tìm kiếm tốt nhất có được hiện tại, nên được cập nhật trong mỗi lần lặp nếu có giải pháp tốt hơn. $\vec{X}$ là vectơ vị trí cá thể tìm kiếm hiện tại.$\vec{A}$ và $\vec{C}$ là các vector hệ số, được tính toán như sau:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (3.5)$$

$$\vec{C} = 2.\vec{r} \quad (3.6)$$

- Phương pháp tấn công lưới bong bóng (giai đoạn khai thác)

Cơ chế bao vây thu hẹp: Hành vi này đạt được bằng cách giảm giá trị của a trong công thức (3.5).

Vị trí cập nhật xoắn ốc:

$$\overrightarrow{X}(t+1) = \vec{D}'.e^{bl}.\cos(2\pi l) + \vec{X}*_{(t+1)} \quad (3.7)$$

WOA Algorithm

1. Khởi tạo một quần thể cá thể tìm kiếm với các vị trí X_i (i=1,2,...,n) Ước tính giá trị thích nghi của mỗi cá thể tìm kiếm

2. X_{best} = cá thể tìm kiếm tốt nhất

3. Cho t = 0

4. while (t < t_{max})

5. for cá thể thứ i : trong Xi

6. Tính các giá trị α, A, C, l và p

7. If (p < 0.5)

8. If (|A| < 1)

9. Cập nhật vị trí của X_i Else

10. Lựa chọn một cá thể Xrand bất kì trong quần thể

11. Cập nhật vị trí của X_i theo phương trình (3.3)

12. Else

13. Cập nhật vị trí của X_i

14. End for

15. Ước tính giá trị thích nghi các mọi cá thể tìm kiếm

16. Cập nhật giá trị cho X_{best}

17. t = t + 1;

18. return Xbest

3. Đề xuất kết hợp thuật toán kết hợp GWO và WOA giải bài toán hệ phương trình phi tuyến

Đề xuất kết hợp thuật toán GWO và WOA cùng với các phương pháp chỉnh sửa khác để tạo ra một thuật toán kết hợp GWOWOA mới nhằm cải thiện hiệu quả

của GWO và đạt kết quả tốt hơn. Thuật toán mới này sẽ thay đổi từ GWO qua các bước sau: đầu tiên, quần thể cho thuật toán sẽ được khởi tạo bằng phương pháp Opposition-Based Learning, một kỹ thuật trong machine learning áp dụng vào các thuật toán tìm kiếm. Trong không gian một chiều, nếu điểm X ở xa vị trí cần tìm, điểm X' đối xứng của X qua trung điểm sẽ gần vị trí cần tìm hơn. Ý tưởng này phức tạp hơn trong không gian đa chiều [12]. Tiếp theo, sửa đổi công thức (3.3) của GWO thành công thức sau:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2}{2} \quad (3.8)$$

GWOWOA Algorithm

1. Khởi tạo một quần thể cá thể tìm kiếm với các vị trí (*i=1,2,...,n*)
2. Ước tính giá trị thích nghi của *mỗi cá thể tìm kiếm*
3. X_α = *cá thể tìm kiếm tốt nhất*
4. X_β = *cá thể tìm kiếm thứ hai*
5. *Cho t = 0*
6. *while (t < tmax)*
7. for cá thể thứ i : trong X_i
8. Tính các giá trị α, A, C và p
9. *If (p < 0.5)*
10. *If (|A| >1)*
11. Cập nhật vị trí của X_i theo phương trình (3.8)
12. Else
13. *Lựa chọn một cá thể Xrand bất kì* trong quần thể
14. Cập nhật vị trí của X_i theo phương trình (3.8)
15. Else
16. Cập nhật giá trị cho X_α, X_β (alpha, beta)
17. t = t + 1;
18. return X_α

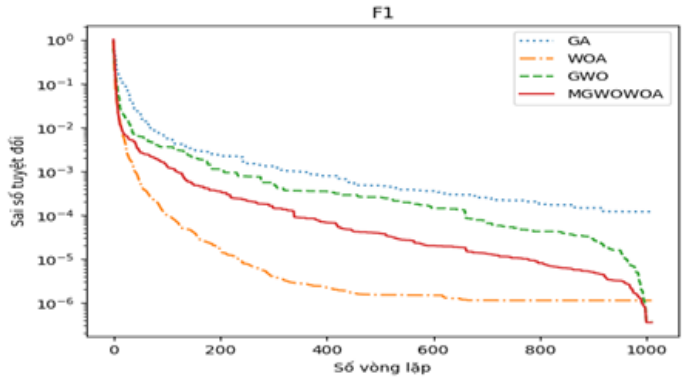
4. Thực nghiệm và đánh giá

Khi áp dụng thuật toán tối ưu hóa vào việc giải các hệ phương trình phi tuyến, chúng sẽ được chuyển thành bài toán tối ưu hóa và giải bằng các phương pháp như GWOWOA mới, GWO tiêu chuẩn, WOA tiêu chuẩn và GA. Thực nghiệm sẽ thực hiện với kích cỡ quần thể 50, qua 1000 vòng lặp và lấy kết quả sau 50 lần thực thi cho mỗi bài toán. Mô phỏng được thực hiện trên máy tính với cấu hình cụ thể. Kết quả và quá trình thực thi sẽ được ghi chép và thể hiện qua bảng và đồ thị, bao gồm kết quả tốt nhất, giá trị trung bình, thời gian trung bình và quá trình hội tụ. Từ đó, đánh giá hiệu quả của GWOWOA so với các thuật toán khác trong giải hệ phương trình phi tuyến. Thuật toán GWOWOA kết hợp GWO và WOA sẽ hoạt động tốt đối với các hệ phương trình phi tuyến. Ứng dụng thuật toán GWOWOA trong bài toán $\boldsymbol{F_1}$ cụ thể:

$$\begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ x_1 - x_2 = 0 \end{cases}$$

Bảng 1. Kết quả thực nghiệm với bài toán F_1

Bài toán	Kết quả	Thuật toán			
		MGWOWOA	GWO	WOA	GA
F_1	Kết quả tốt nhất	1.28E-10	1.24E-09	3.33E-11	8.15E-08
	Kết quả trung bình	1.04E-08	1.61E-08	3.65E-07	3.98E-06
	Thời gian thực thi	1.749s	1.999s	1.775s	2.208s



Hình 1. Sơ đồ thể hiện sự hội tụ của các giải thuật trong bài toán F_1

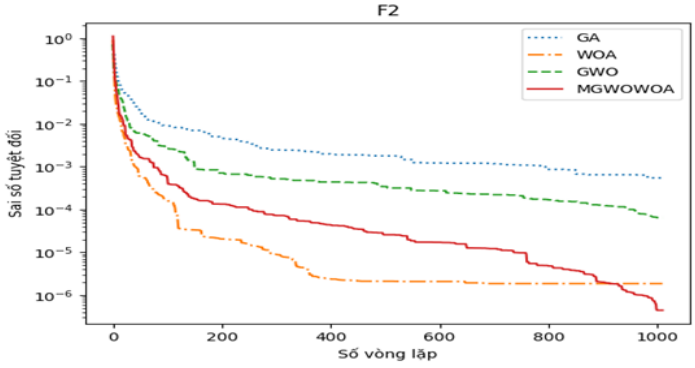
Các giải thuật sẽ được tiến hành với kích cỡ quần thể là 50 và qua 1000 vòng lặp và kết quả được lấy sau 50 lần thực thi khác nhau cho từng bài toán hệ phương trình phi tuyến một. Việc mô phỏng thực được thực hiện trên máy tính có bộ xử lý là Intel® Core™ i5-8250U CPU @ 1.60GHz (8 CPUs) ~1.8GHz, với bộ nhớ 8GB RAM .Về thời gian thực thi, hiệu năng của thuật toán GWOWOA trong bài toán F_1 mang tính cạnh tranh so với các thuật toán GWO và WOA. Thuật toán GWOWOA cho kết quả (với thời gian là 1.749s) nhanh hơn cả 2 thuật toán GWO và WOA. Ngoài ra, cho giá trị trung bình là 1.04 x10^{-8} tốt hơn các thuật toán còn lại.

*** Ứng dụng thuật toán** GWOWOA trong bài toán F_2 cụ thể

$$\begin{cases} \boldsymbol{sin(5\pi x_1) - x_2 = 0} \\ \boldsymbol{x_1 - x_2 = 0} \end{cases}$$

Bảng 2. Kết quả thực nghiệm với bài toán F_2

Bài toán	Kết quả	Thuật toán			
		MGWOWOA	GWO	WOA	GA
F_2	Kết quả tốt nhất	0.00E-00	0.00E-00	0.00E-00	8.89E-07
	Kết quả trung bình	1.48E-08	2.16E-06	6.20E-08	1.83E-05
	Thời gian thực thi	2.382s	3.557s	2.837s	3.856s



Hình 2. Sơ đồ thể hiện sự hội tụ của các giải thuật trong bài toán F_2

Các giải thuật sẽ được tiến hành với kích cỡ quần thể là 50 và qua 1000 vòng lặp và kết quả được lấy sau 50 lần thực thi khác nhau cho từng bài toán hệ phương trình phi tuyến một. Việc mô phỏng thực được thực hiện trên máy tính có bộ xử lý là Intel® Core™ i5-8250U CPU @ 1.60GHz (8 CPUs) ~1.8GHz, với bộ nhớ 8GB RAM. Về thời gian thực thi, hiệu năng của thuật toán GWOWOA trong bài toán F_2 mang tính cạnh tranh so với các thuật toán GWO và WOA cụ thể: thời gian xử lý nhanh và kết quả tương đồng với 2 thuật toán GWO và WOA. Thuật toán GWOWOA cho kết quả (với thời gian là 2.382s) nhanh hơn cả 3 thuật toán GWO, WOA và GA. Ngoài ra, cho giá trị trung bình Mean (1.48E-08) tốt hơn các thuật toán còn lại.

Phân tích và đánh giá kết quả:

Với 2 bài toán thử nghiệm trên, ta có thể thấy rằng thuật toán mới GWOWOA đem lại kết quả triển vọng hơn so với các thuật toán còn lại trong đa số bài toán. Về thời gian thực thi, hiệu năng của thuật toán rất cạnh tranh so với các thuật toán GWO và WOA, không bao giờ triển khai lâu hơn cả 2 thuật toán và đôi khi lại nhanh hơn 2 thuật toán trên, điều này được thể hiện ở bài toán $\boldsymbol{F_1}$, $\boldsymbol{F_2}$. Qua đó thuật toán này có thể xem là rất có tiềm năng trong việc áp dụng vào giải các hệ phương trình phi tuyến.

5. Kết luận

Bài báo này tổng hợp và đánh giá các thuật toán tối ưu hóa như GWO, GA, PSO, và WOA, cùng với lý thuyết giải các hệ phương trình phi tuyến, với sự đánh giá đặc biệt về hiệu quả của thuật toán GWOWOA trong việc giải các hệ phương trình phi tuyến. Tuy nhiên, bài báo gặp phải một số hạn chế trong quá trình thử nghiệm và đòi hỏi thêm nghiên cứu để khắc phục, đặc biệt là sự thiếu đa dạng trong các bài toán thực nghiệm. Trong tương lai, hướng phát triển có thể tập trung vào việc cải tiến các phương pháp tối ưu hóa nhằm giải quyết hiệu quả những bài toán này.

TÀI LIỆU THAM KHẢO

[1]. C. Grosan and A. Abraham, “A New Approach for Solving Nonlinear Equations Systems", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 38, no. 3, pp. 698-714, May 2008.

[2]. D. Jitkongchuen, "A hybrid differential evolution with grey wolf optimizer for continuous global optimization," International Conference on nformation Technology and Electrical Engineering (ICITEE), 2015, pp. 51-54. doi: 10.1109/ICITEED.2015.7408911.

[3]. Chao Lu, Shengqiang Xiao, Xinyu Li, and Liang Gao, “An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production,” Advances in Engineering Software, vol. 99, pp. 161-176, 2016.

[4]. Phan Quốc Khánh, “Vận Trù Học”, Nhà xuất bản Giáo Dục , 2006.

[5]. S. Mirjalili, S. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[6]. P. Erdogmus, “A New Solution Approach for Non-Linear Equation Systems with Grey Wolf Optimizer,” Sakarya University Journal of Computer and Information Sciences, vol. 1, no. 3, pp. 1-11, 2019.

[7]. M. Esmaelian, M. Tavana, F. J. Santos-Arteaga, and M. Vali, “A novel genetic algorithm based method for solving continuous nonlinear optimization problems through subdividing and labeling,” Measurement, vol. 115, pp. 27-38, 2018.

[8]. G. Joshi and M. B. Krishna, "Solving system of non-linear equations using Genetic Algorithm," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 1302-1308.

[9]. Z.-M. Gao and J. Zhao, “An Improved Grey Wolf Optimization Algorithm with Variable Weights,” Computational Intelligence and Neuroscience, vol. 2019, pp. 1-13, 2019.

[10]. S. Qin, S. Zeng, W. Dong, and X. Li, "Nonlinear equation systems solved by many-objective Hype," 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, 2015, pp. 2691-2696.

[11]. W. Song, Y. Wang, H. X. Li, and Z. Cai, "Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization," IEEE Transactions on Evolutionary Computation, vol. 19, no. 3, pp. 414-431, June 2015.

[12]. X.-S.Yang,Nature-Inspired Optimization Algorithms (Second Edition), Academic Press/Elsevier, ISBN: 9780128219867, Sep. 2020.

[13]. Hoàng Thị Dịu, “Một số phương pháp giải hệ phương trình và bất phương trình đại số,” luận văn thạc sĩ, chuyên ngành toán học, Trường đại học Khoa học Tự nhiên, Đại học Quốc Gia Hà Nội, 2014.

[14]. Lê Tấn Long, “Ứng dụng thuật toán tối ưu hóa GWO cải tiến vào việc tìm lời giải cho hệ phương trình phi tuyến,” Luận văn thạc sĩ chuyên ngành Khoa học máy tính, Trường Đại học Sài Gòn, 2021.

Ngày nhận bài: 07/6/2024 Ngày chấp nhận đăng: 08/7/2024