

Thiết kế hệ điều khiển mờ tối ưu cho Robot tự hành sử dụng giải thuật Sói Xám

Nguyễn Hồng Ngọc¹

¹⁻² Viện công nghệ thông tin & Truyền thông, Trường Đại học Bách Khoa Hà Nội,

Giảng viên hướng dẫn: Phạm Văn Hải

TÓM TẮT— Với sự phát triển của công nghệ, hàng ngày, càng nhiều phương pháp và kỹ thuật tính toán mới lại xuất hiện cho phép chúng ta giải quyết các vấn đề thuộc nhiều lĩnh vực ngành nghề khác nhau, như y tế, kỹ thuật và công nghiệp. Tối ưu hóa là một vấn đề quan trọng trong ngành công nghiệp này, với mục tiêu chính là tìm ra lời giải tốt nhất có thể cho bài toán cần giải quyết. Trong bài báo này, nhóm nghiên cứu đề xuất sử dụng giải thuật Tối ưu đàn sói xám (GWO) - một giải thuật xấp xỉ dựa trên đặc tính săn mồi trong tự nhiên và hệ thống cây phân cấp chỉ huy (con đầu đàn) của loài sói xám. Cùng với đó, nhóm đưa ra các phân tích, đánh giá và giải thích áp dụng thuật toán GWO cho bài toán tối ưu hóa bộ điều khiển mờ cho robot tự hành.

Từ khóa— Thuật toán sói xám GWO, Robot tự hành, Điều khiển mờ, Tối ưu hóa, Hệ mờ, Thuật toán dựa trên sinh học.

I. GIỚI THIỆU

Các phương pháp tính toán dựa trên sinh học thường lấy ý tưởng từ các hành vi và quá trình trong tự nhiên cũng như xã hội để giải quyết các vấn đề phức tạp. Các thuật toán này thường mô phỏng các hành vi trong tự nhiên để thiết kế ra các phương pháp xấp xỉ gần đúng và không xác định. Mục đích của các thuật toán này chủ yếu là để tìm kiếm trong không gian lời giải, học đặc trưng hoặc phản ứng lại đầu vào. [34]

Đặc tính của tối ưu hóa là tìm ra lời giải tốt nhất cho bất cứ vấn đề nào trong tính toán, dù là tìm giá trị nhỏ nhất hay lớn nhất. Với sự phức tạp và đa dạng của bài toán, chúng ta cần đến những giải thuật dựa trên sinh học mà có khả năng mô phỏng lại các quá trình tự nhiên nhằm tìm kiếm điều kiện tốt nhất. Các giải thuật dựa trên tự nhiên thường có những pha nhất định như: đầu vào (các biến số), quá trình phát triển (mức độ thích nghi) và đầu ra (chi phí) [13]. Ở các giải thuật này, thông thường mỗi lời giải có thể của bài toán được mô phỏng như một cá thể tìm kiếm lời giải. Lời giải đạt được thông qua việc tiến hóa tập cá thể hay tập các lời giải sao cho tốt dần qua từng thế hệ cho đến khi hội tụ. Lời giải tốt nhất được tìm ra trong quá trình thực hiện thuật toán được chọn làm đầu ra của bài toán. Hoạt động và hiệu quả của thuật toán phụ thuộc rất nhiều vào việc chọn lựa phương pháp tiến hóa tập lời giải. Trên hết, do tính ngẫu nhiên của quá trình tiến hóa, các thuật toán này cho kết quả khác nhau trong những lần chạy khác nhau và giá trị thu được tiến gần đúng đến giá trị chính xác. Vì thế các thuật toán này được xếp vào lớp giải thuật tối ưu hóa xấp xỉ gần đúng.

Có rất nhiều các giải thuật dựa trên sinh học có thể giúp chúng ta giải quyết các vấn đề khó, ví dụ như: giải thuật thụ phấn cho hoa [35], giải thuật đàn dơi [36], giải thuật đom đóm [32], giải thuật chim cockoo [33], giải thuật vi khuẩn tìm kiếm thức ăn [1], giải thuật đàn ong [14], giải thuật đàn kiến [28], giải

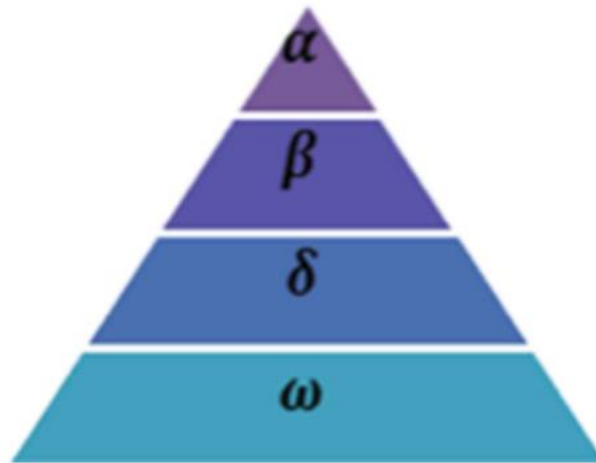
thuật bầy đàn PSO [15], giải thuật di truyền [10]. Đây là những giải thuật được sử dụng nhiều nhất trong lĩnh vực này.

Nghiên cứu này tập trung vào việc tìm ra một hướng giải quyết mới trong tối ưu hóa điều khiển mờ sử dụng giải thuật đàn sói xám để nâng cao hiệu năng của thuật toán. Ở đây, bài toán được đưa vào để thử nghiệm mô hình là bài toán tối ưu hóa điều khiển mờ cho robot tự hành. Mô hình được sử dụng trong bài báo này là mô hình robot unicycling, đã được nghiên cứu áp dụng từ nhiều bài báo liên quan trong lĩnh vực điều khiển robot. Một trong những bài báo đầu tiên đề xuất mô hình toán học của robot unicycle là nghiên cứu [40] sử dụng phản hồi saturation để điều khiển. Sau đó mô hình này được đưa vào điều khiển bằng logic mờ với tối ưu hóa tập mờ sử dụng các giải thuật tìm kiếm xấp xỉ: [3] sử dụng tối ưu hóa bầy đàn để giải; [6] sử dụng giải thuật thụ phấn cho hoa để giải; [8] sử dụng giải thuật đàn kiến để giải và [16] sử dụng giải thuật di truyền. Nghiên cứu [2] đã thống kê lại và so sánh giữa các giải thuật tiến hóa khác nhau cùng giải trên bài toán này. Đây là bài báo đầu tiên sử dụng giải thuật đàn sói xám để giải bài toán tối ưu hóa điều khiển mờ cho robot tự hành unicycling.

II. THUẬT TOÁN BẦY SÓI XÁM GWO

1. Tối ưu bầy sói xám

Seyedali Mirjalili đề xuất giải thuật GWO vào năm 2014. Về cơ bản, giải thuật GWO mô phỏng lại tập tính săn mồi của bầy sói xám. Nói cách khác, GWO áp dụng tập tính phân cấp chỉ huy điều khiển cuộc săn mồi và cơ chế săn mồi của sói xám vào tối ưu hóa hàm. Có thể nói rằng, trong bầy sói, thường sẽ có 4 loại sói xám khác nhau: alpha, beta, delta và omega phân cấp theo kim tự tháp như sau: [22]



Hình 1: Cây phân cấp đầu đàn theo kim tự tháp

Quá trình săn mồi của GWO được tổng quát như sau: [23]:

- Định vị, đuổi và tiếp cận con mồi
- Vây hãm, khép vòng và bao vây con mồi cho đến khi nó ngừng di chuyển
- Tấn công con mồi

Trong phần này, một số khái niệm sẽ được nêu rõ để đem lại nhìn toàn cảnh tốt hơn về giải thuật được đề xuất.

2. Bao vây con mồi

Loài sói thường có đặc tính bao vây con mồi trong cuộc săn của chúng. Để có thể mô hình đặc tính này dưới dạng toán học, các phương trình (1) và (2) được đưa ra như sau:

$$D = |CX_p(t) - X(t)| \quad (1)$$

$$X(t+1) = X_p(t) - AD \quad (2)$$

Trong đó:

- t : số chỉ mục của vòng lặp hiện tại
- A, C : các tham số
- X_p : vị trí của con mồi
- X : vị trí của sói xám
- D : đại diện cho khoảng cách

Các tham số A và C được tính theo công thức (3) và (4) sau:

$$A = 2ar_1 - a \quad (3)$$

$$C = 2r_2 \quad (4)$$

Trong đó, a là một số giảm tuyến tính từ 2 đến 0 qua quá trình các vòng lặp, r_1 và r_2 là các số ngẫu nhiên trong khoảng $[0, 1]$

3. Săn mồi

Loài sói xám có bản năng tự nhiên giúp cảm nhận vị trí con mồi và vây bắt nó. Con sói Alpha đầu đàn đôi khi đảm nhiệm công việc chỉ huy cuộc săn, đôi khi lại tiến hóa con sói Beta và Delta. Để có thể mô phỏng toán học điều này, chúng ta coi con sói Alpha (hay tương ứng với lời giải tốt nhất), Beta và Delta có hiểu biết tốt hơn về những vị trí có thể của con mồi, theo như phương trình (5) (6) và (7). Vì thế, chúng ta giữ lại ba lời giải tốt nhất cho đến hiện tại và bắt các agent khác (các con sói Omega) cập nhật vị trí của chúng dựa trên vị trí của ba con đầu đàn.

Nói cách khác, các phương trình sau đây được đề xuất:

$$D_\alpha = |C_1 X_\alpha - X| \quad D_\beta = |C_2 X_\beta - X| \quad D_\delta = |C_3 X_\delta - X| \quad (5)$$

$$X_1 = X_\alpha - A_1(D_\alpha) \quad X_2 = X_\beta - A_2(D_\beta) \quad X_3 = X_\delta - A_3(D_\delta) \quad (6)$$

$$X(t+1) = \frac{x_1 + x_2 + x_3}{3} \quad (7)$$

Về cơ bản, vị trí mới của các con sói omega sẽ nằm ngẫu nhiên xung quanh ba con sói đầu đàn, qua quá trình thuật toán thì phạm vi được thu hẹp lại dần cho đến khi hội tụ.

4. Tấn công con mồi

Bầy sói xám tấn công con mồi khi con mồi ngừng di chuyển. Để mô hình toán học điều này, giá trị của tham số a sẽ giảm dần theo thời gian thực hiện thuật toán. Khi mà giá trị của tham số A nằm ngoài khoảng $[-1, 1]$, giai đoạn này là giai đoạn tìm kiếm con mồi hay giai đoạn khám phá. Khi mà giá trị của tham số A nằm trong khoảng $[-1, 1]$, con mồi sẽ bị tấn công. Giai đoạn này còn được gọi là giai đoạn khai phá.

Mã giả của thuật toán GWO như sau:

1	Khởi tạo quần thể sói $X_i (i = 1, 2, \dots, n)$
2	Khởi tạo các biến số a, A, C
3	Tính giá trị fitness của mỗi con sói
4	Alpha = Con sói tốt nhất
5	Beta = Con sói tốt thứ 2
6	Delta = Con sói tốt thứ 3
7	While (t < Số lượng thế hệ tối đa):
8	For mỗi con sói:
9	Cập nhật vị trí theo phương trình (7)
10	End for
11	Cập nhật a, A, C
12	Tính toán giá trị fitness của các con sói
13	Cập nhật lại Alpha, Beta, Delta
14	t ++
15	End while
16	Trả về Alpha là lời giải

III. MÔ HÌNH BÀI TOÁN

1. Logic mờ

Logic mờ được đề xuất bởi LA Zadeh vào năm 1965, từ trường đại học California, Berkeley, dựa trên một công trình về lý thuyết tập mờ. Lý thuyết tập mờ cho phép biểu diễn tri thức của con người, xem xét các thông tin không chính xác bằng cách kết hợp các luật hoặc biến ngôn ngữ, ví dụ như: "Rất thấp", "Thấp", "Trung bình", "Cao" và "Rất cao". [37-39]

Logic rõ truyền thống chỉ chấp nhận sử dụng hai giá trị đó là hoàn toàn đúng và hoàn toàn sai. Ví dụ, câu "Ngày mai trời sẽ ấm" phải mang giá trị đúng hoặc sai. Tuy nhiên, thông tin mà con người sử dụng luôn mang một mức độ không chắc chắn nhất định. Trong logic mờ cho phép xuất hiện giá trị mức độ đúng nằm giữa 0 và 1. Ví dụ, câu "Tổng thống Mexico thì già" có thể có mức độ chính xác là 0.7. Logic mờ sẽ cố gắng đo đạc mức độ đó và giúp máy tính có khả năng hiểu được thông tin này.

2. Robot tự hành

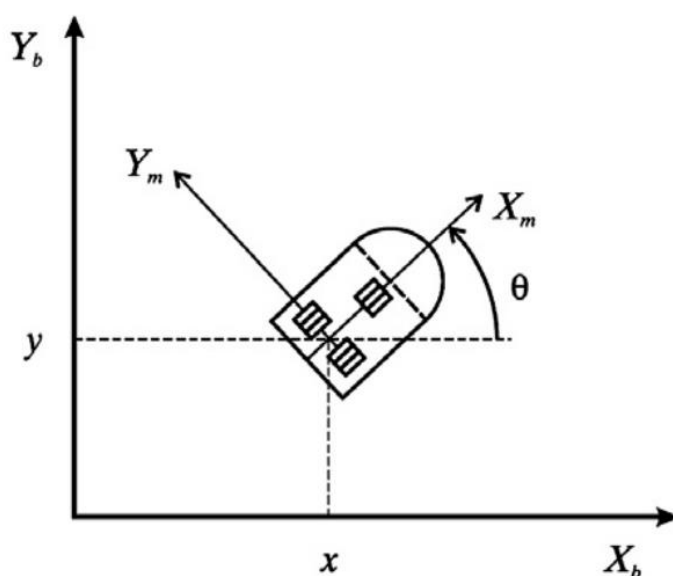
Robot tự hành là những hệ thống có khả năng tương tác với mức độ tự động hóa cao trong giới hạn của môi trường xung quanh nó. Robot có khả năng chuyển vào bất cứ môi trường nào tùy thuộc vào chương trình phần mềm được lập trình. Điều này dẫn đến ứng dụng to lớn của robot tự hành trong: khám phá vũ trụ, thăm dò vùng ô nhiễm, hoặc các công việc nguy hiểm vượt ngoài khả năng của con người. [7, 16].

Robot trong công nghiệp đã phát triển một cách mạnh mẽ, từ những thiết bị văn phòng phẩm cho đến những nền tảng di động cấp cao, giúp cải thiện nâng cao chất lượng công việc. Đây là lý do ngành công nghiệp robot đã đạt được nhiều thành công lớn trong một thế giới mà công nghiệp hóa phát triển. Hiện tại,

những cánh tay robot được thiết kế có độ chính xác và tốc độ cao, dùng để thực hiện nhiều công việc khác nhau, như di chuyển vật nặng hoặc phân loại vật liệu.

Robot di động tự hành có khả năng nhận diện điều hướng trong môi trường thực tế để tìm đường đi. Một trong những tiến bộ vượt bậc đó là sự tự động hóa. Các loại robot tự hành thường yêu cầu rất ít dữ liệu đầu vào nhập từ người dùng hay chỉ dẫn, đem lại khả năng làm việc cao hơn. Một số tiến bộ được phát kiến từ các hệ thống thông minh như hệ neural network, giải thuật lấy ý tưởng từ sinh học, ... Sau đó, robot được trang bị các cảm biến giúp chúng có khả năng thu nhận các biến động môi trường trong thời gian thực. Robot tự hành có tiềm năng công nghiệp to lớn và dễ dàng được áp dụng vào triển khai trong nhiều ngành nghề khác nhau.

3. Mô hình robot tự hành unicycle



Hình 2: Mô hình robot unicycle

Robot Unicycle gồm 2 bánh xe được lắp cố định vào cùng 1 trục và 1 bánh trước tự do di chuyển (quay). Robot Unicycle là robot tự hành, thuộc dạng xe phương tiện có bánh, có khả năng di chuyển trong môi trường cố định hoặc biến động. Phần thân robot có cấu trúc đối xứng trục và trọng tâm của robot nằm ở trọng tâm hình học của phần thân. Hai bánh sau của robot là bánh lái, được gắn cố định vào trục và bánh trước của robot là bánh bị động, được thả tự do để ngăn không cho robot bị chúi đầu về phía trước và đổ khi di chuyển. Trong mô hình được trình bày sau đây, chúng ta giả thiết rằng chuyển động bánh trước của robot sẽ được bỏ qua và không có tác động quá nhiều. Sự vận hành của robot tự hành được mô tả qua phương trình dưới:

$$M(q)\dot{\vartheta} + C(q, \dot{q})\dot{\vartheta} + D\vartheta = \tau + F_{ext}(t),$$
$$\dot{q} = \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{\vartheta},$$

Trong đó:

$q = (x, y, \theta)^T$	Vector tọa độ của robot
$\vartheta = (v, w)^T$	Vector vận tốc tuyến tính và vận tốc góc
$\tau = (\tau_1, \tau_2)$	Vector lực tác dụng lên 2 bánh xe, lần lượt phải và trái
$F_{ext} \in \mathbb{R}^2$	Ngẫu số phân phối đều trong khoảng bị chặn
$M(q) \in \mathbb{R}^{2 \times 2}$	Ma trận quán tính xác định dương
$C(q, \dot{q})\vartheta$	Vector hướng tâm và lực Coriolis
$D \in \mathbb{R}$	Ma trận giảm xóc chéo xác định dương

Phương trình trên mô tả động học của robot unicycle, với (x,y) là tọa độ của robot trong hệ Đề các, và θ là góc giữa vector hướng của robot với phương nằm ngang. Ngoài ra, hệ còn có một điều kiện đó là:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0,$$

Điều kiện này nhằm đảm bảo bánh xe không bị trượt và ngăn cho robot không di chuyển ngang. Mục đích của điều khiển là phải thiết kế một hệ mờ cho đầu ra τ sao cho đảm bảo:

$$\lim_{\tau \rightarrow \infty} \|qd(t) - q(t)\| = 0$$

Trong đó qd là quỹ đạo mong muốn và q là quỹ đạo thực tế robot tự hành.

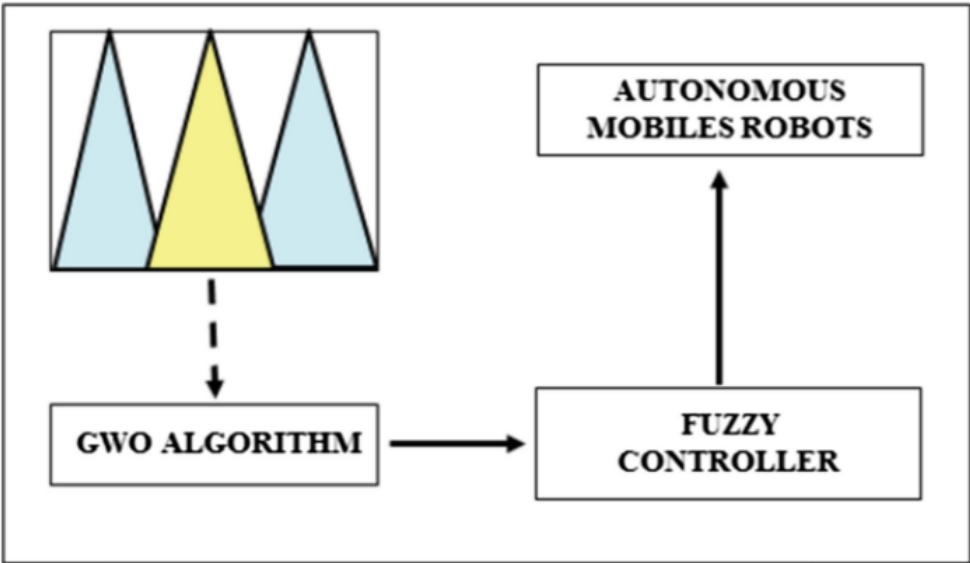
IV. PHƯƠNG PHÁP ĐỀ XUẤT

1. Phương pháp

Một hệ thống được tạo ra để tối ưu hóa bộ điều khiển mờ cho robot tự hành unicycle sử dụng giải thuật tối ưu sói xám để cho phép robot đi theo đúng quỹ đạo đã định.

Trước tiên, chúng ta đưa ra các tham số cần được tối ưu trong các hàm độ thuộc (membership function). Sau đó, giải thuật GWO được áp dụng vào hệ mờ để tối ưu hóa các tham số này. Sau khi chạy GWO, chúng ta thu được hệ với các tham số đã được tối ưu. Cuối cùng, đưa tập mờ vào thử nghiệm và mô phỏng, theo dõi sự chênh lệch của đường đi thực tế với đường đi mong muốn.

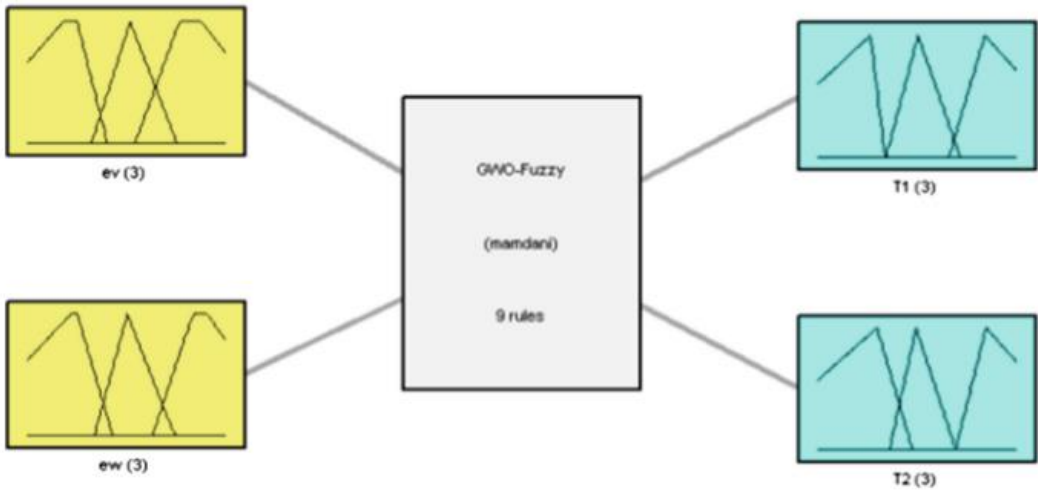
Hình sau đây mô tả hệ thống tối ưu được thiết kế



Hình 3. Hệ thống tối ưu tập mờ

2. Bộ điều khiển mờ

Các luật của bộ điều khiển mờ mô tả mối quan hệ giữa các biến ngôn ngữ dựa trên các hàm độ thuộc , xem xét đến các giá trị đầu vào và đầu ra. [2,6,8,27]



Hình 4. Hệ điều khiển mờ

Một hệ thống mờ Mamdani được thiết kế bao gồm đầu vào là hai giá trị: error của vận tốc tuyến tính ev và error của vận tốc góc ew . Hai giá trị này được chuẩn hóa về khoảng $[-1, 1]$, bao gồm ba hàm thuộc: 2 hàm độ thuộc hình thang ở 2 đầu và 1 hàm độ thuộc tam giác ở giữa. Hệ mờ cũng bao gồm hai giá trị đầu ra là T1 - lực tác dụng lên bánh xe phải và T2 - lực tác dụng lên bánh xe trái. Giá trị cũng được chuẩn hóa về đoạn $[-1, 1]$ và bao gồm ba hàm thuộc tam giác. Hệ này được mô tả trong hình 4 [5,7,29,30].

Bộ điều khiển mờ này bao gồm 9 luật if-then, được mô tả trong hình 5. [2,24,31].

e_v / e_w	N	C	P
N	N / N	N / C	N / P
C	C / N	C / C	C / P
P	P / N	P / C	P / P

Hình 5. Luật mờ

Theo như bào báo tham chiếu gốc, bộ tham số được giới hạn trong khoảng tìm kiếm sau đây:

Bảng 1. Miền tìm kiếm bộ tham số tập mờ

MF Type	Point	Minimum Value	Maximum Value
Trapezoidal	a	-50	-50
	b	-50	-50
	c	-15	-5.05
	d	-1.5	-0.5
Trapezoidal	a	-5	-1.75
	b	0	0
	c	1.75	5
Trapezoidal	a	0.5	1.5
	b	5.05	15
	c	50	50
	d	50	50

3. Tối ưu hóa bằng giải thuật đàn sói xám

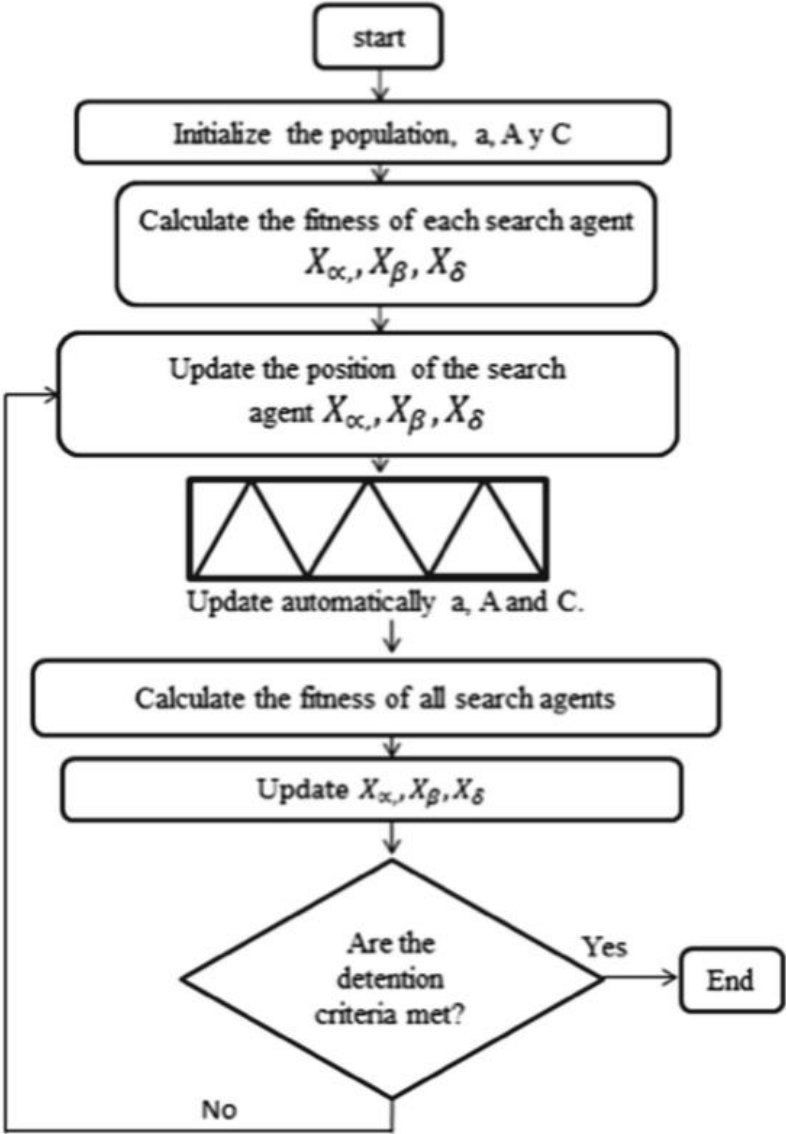
Về cơ bản, nghiên cứu này tập trung vào tối ưu tập mờ để đạt được kết quả tốt nhất. Hình 6 thể hiện mô hình thuật toán sử dụng - giải thuật đàn sói xám GWO.

Giống như các thuật toán tối ưu hóa theo quần thể, các bước chính bao gồm:

- Khởi tạo: Sinh ngẫu nhiên tập các agent trong quần thể
- Vòng lặp tiến hóa: Liên tục lặp lại quá trình tiến hóa quần thể để cải thiện lời giải
- Kết thúc: Kiểm tra điều kiện dừng và kết thúc

Với GWO, các bước và quá trình tương tự cũng được diễn ra. Trong GWO, quá trình tiến hóa được thể hiện thông qua việc di chuyển của các con sói Omega theo ba con chỉ huy đầu đàn là Alpha, Beta và Delta. Ba con sói này có kinh nghiệm tốt hơn các con khác và sẽ làm nhiệm vụ định vị vị trí của "con mồi" để chỉ đạo cuộc đi săn.

Chi tiết thuật toán được mô tả trong lược đồ dưới:



Hình 6. Sơ đồ giải thuật GWO sử dụng

v. KẾT QUẢ THỬ NGHIỆM

1. Thiết lập tham số

Bộ điều khiển mờ được thiết lập như trong bài báo tham chiếu gốc. Quỹ đạo mong muốn của Robot được mô tả theo hàm sau: (hàm của vector vận tốc)

$$\vartheta_d(t) = \begin{cases} v_d(t) = 0.2(1 - \exp(-t)) \\ w_d(t) = 0.4 \sin(0.5 t) \end{cases}$$

Các giá trị tham số được tính như sau:

$$M(q) = \begin{bmatrix} 0.3749 & -0.0202 \\ -0.0202 & 0.3739 \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0.1350\dot{\theta} \\ -0.1350\dot{\theta} & 0 \end{bmatrix},$$

$$\text{and } D = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Điều kiện khởi đầu của xe tự hành:

$$q(0) = (0.1, 0.1, 0)^T \text{ and } \dot{q}(0) = 0 \in \mathbb{R}^2$$

Các thử nghiệm khác nhau được xây dựng để kiểm thử giải thuật GWO. Các tham số được thiết lập như sau:

- Số lượng agent: 50
- Số vòng lặp: 500

Giá trị error được tính thông qua phương sai. Theo đó, Mean Square Error bằng:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Giá trị này cũng được sử dụng để làm fitness cho giải thuật GWO. Theo đó, GWO sẽ tối ưu tham số theo việc tìm ra bộ đầu vào sao cho giá trị fitness nhỏ nhất.

2. Kết quả

Bảng 2 cho thấy kết quả của 15 thực nghiệm. Kết quả đạt được tốt nhất tại bước thứ 15 với 5000 vòng lặp, 50 agent, đạt giá trị error tốt nhất là 0.0000442498.

Bảng 1. Kết quả GWO khi thay đổi tham số

No.	Iterations	Agents	Error
1	500	55	0.000741061
2	400	50	7.29585E-05
3	1000	60	0.003882946
4	600	45	0.064997856
5	450	65	0.018010558
6	500	45	0.000626393
7	800	40	0.021905084
8	550	56	0.019565891
9	4000	60	3.19E-04
10	3000	70	0.0021119
11	2000	75	0.001740619
12	1000	65	0.008165245
13	800	55	0.000436007
14	700	60	0.000417135
15	5000	50	0.42498E-05

VI. DEMO THUẬT TOÁN

1. Môi trường Demo

Các thông tin về môi trường demo code như sau:

- Ngôn ngữ sử dụng: Python
- Môi trường cài đặt thuật toán: Jupyter Notebook
- Môi trường phần cứng cài đặt: Laptop - CPU: Intel Core i7 4720HQ, Ram 8GB

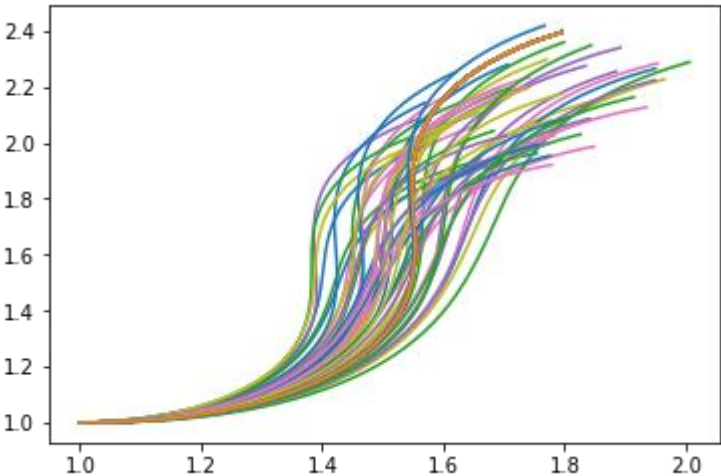
Tham số sử dụng:

- Số lượng Agent: 20 ~ 100
- Số vòng lặp: 10 ~ 20

2. Phân tích và kết quả Demo

Mỗi agent trong đàn sói xám tương ứng với một bộ tham số hàm thuộc của tập mờ được sử dụng trong bộ điều khiển mờ. Với mỗi bộ điều khiển mờ sử dụng tham số khác nhau, robot unicycle sẽ di chuyển theo một quỹ đạo khác nhau. Mục tiêu của tối ưu hóa bầy sói xám là tìm kiếm bộ tham số cho đường đi giống nhất so với quỹ đạo mong muốn cho trước.

Ví dụ, hình ảnh khởi tạo quần thể agent: Đường màu cam ở giữa là đường đi mong muốn (desired trajectory). Các đường khác là đường đi được sinh ra từ việc sử dụng các tham số hệ mờ tương ứng với các agent trong quần thể sói xám.



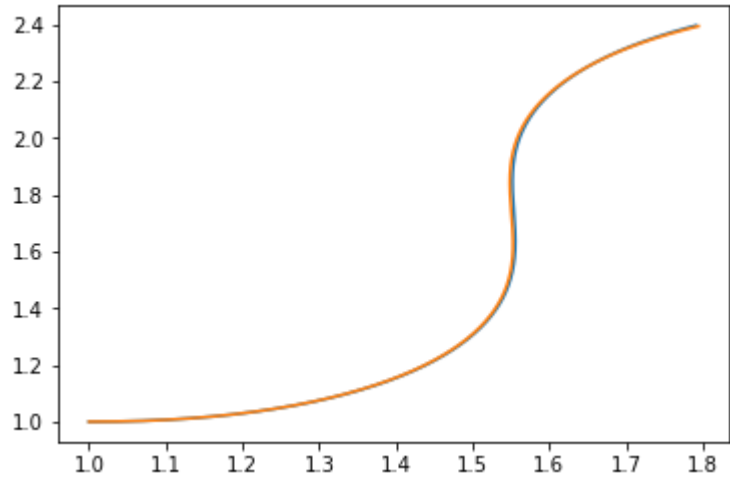
Hình 7. Khởi tạo quần thể

Chạy thuật toán: Giá trị error (MSE) được sử dụng để làm hàm thích nghi của GWO. Trong quá trình chạy thuật toán, giá trị này giảm dần tương ứng với quần thể được tiến hóa dần:

```
Init: Fit = 0.001136812702551211
Gen 0 : Fit = 0.00034467522952670896
Gen 1 : Fit = 0.0001910874731622486
Gen 2 : Fit = 0.00016649240129609105
Gen 3 : Fit = 0.0001633328420625831
Gen 4 : Fit = 0.00013078861516168687
Gen 5 : Fit = 0.00010302685659652039
Gen 6 : Fit = 9.177216856988743e-05
Gen 7 : Fit = 8.197693304036405e-05
Gen 8 : Fit = 7.354627054718069e-05
Gen 9 : Fit = 7.354627054718069e-05
Gen 10 : Fit = 7.265897565686343e-05
```

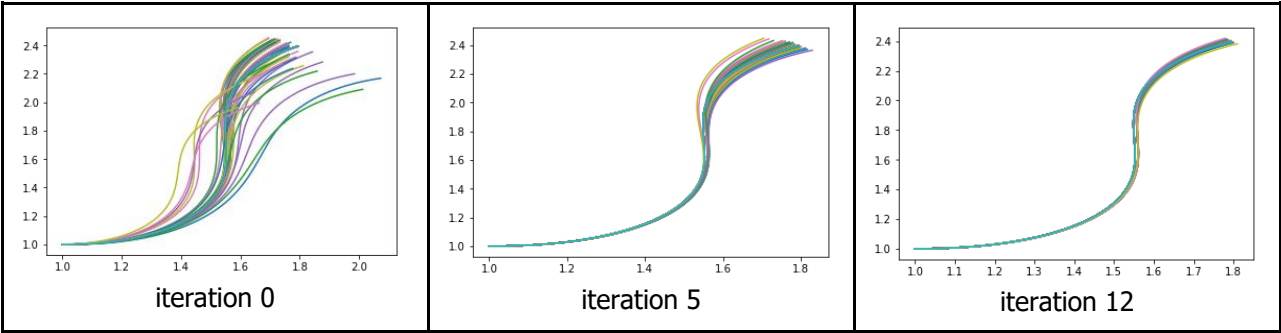
Gen 11 : Fit = 7.226991609014131e-05
Gen 12 : Fit = 7.012802836910306e-05
Gen 13 : Fit = 7.012802836910306e-05
Gen 14 : Fit = 7.012802836910306e-05
Gen 15 : Fit = 7.012802836910306e-05
Gen 16 : Fit = 7.012802836910306e-05
Gen 17 : Fit = 7.012802836910306e-05
Gen 18 : Fit = 7.012802836910306e-05
Gen 19 : Fit = 7.012802836910306e-05

Kết quả cuối cùng: đường đi thực tế so với đường đi mong muốn:



Hình 8. Kết quả sau khi tối ưu

Quá trình hội tụ của bầy sói xám:



Hình 9. Quá trình hội tụ

Nhận xét: Quá trình hoạt động của thuật toán GWO giống như dự đoán. Giá trị MSE thu được là khá nhỏ chứng tỏ kết quả thu được khá chính xác.

VII. KẾT LUẬN

Trong bài báo này, tác giả đã tiến hành đánh giá sâu hơn về hiệu năng của thuật toán tối ưu hóa đàn sói xám GWO. Những thực nghiệm khác nhau đã được chạy với hệ mờ điều khiển robot tự hành, quá trình tối ưu hóa đem lại kết quả khá tốt. Với 15 bộ dữ liệu thực nghiệm chạy thuật toán GWO, có thể nói rằng, kết quả thu được là ổn. Trong tương lai, nhóm nghiên cứu sẽ thực hiện nhiều thử nghiệm hơn để đánh giá và so

sánh với các giải thuật metaheuristic khác dựa trên thống kê số liệu. Ngoài ra, nhóm sẽ áp dụng thêm cả logic mờ loại 2 giống như trong [9,11,12,17,25]. Nhóm cũng sẽ cố gắng ứng dụng phương án đề xuất trên cả các lĩnh vực khác như [4,18-21]

Đánh giá riêng của tôi, nhóm nghiên cứu viết bài báo còn khá thiếu rõ ràng về mô hình bài toán, chưa nêu được rõ các thiết lập chạy, gây khó hiểu khá nhiều. Thuật toán chạy trên thực nghiệm cũng không có nhiều ý nghĩa lắm, các so sánh đưa ra là không công bằng. Nhìn chung, chất lượng của nghiên cứu này chỉ ở tầm trung.

VIII. TÀI LIỆU THAM KHẢO

- [1]. Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. *Found. Comput. Intell.* 3, 23–55 (2009)
- [2]. Soria: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile (2012)
- [3]. Abdalla, T.Y., Abdulkareem, A.: A PSO optimized fuzzy control scheme for mobile robot path tracking. *Int. J. Comput. Appl.* 76(2), 11–17 (2013)
- [4]. Aguilar, L., Melin, P., Castillo, O.: Intelligent control of a stepping motor drive using a hybrid neuro-fuzzy ANFIS approach. *Appl. Soft Comput.* 3(3), 209–219 (2003)
- [5]. Astudillo, L., Melin, P., Castillo, O.: Chemical optimization algorithm for fuzzy controller design (2014)
- [6]. Carvajal, O., Castillo, O., Soria, J.: Optimization of membership function parameters for fuzzy controllers of an autonomous mobile robot using the flower pollination algorithm. *J. Autom. Mob. Robot. Intell. Syst* 12(1), 44–49 (2018)
- [7]. Castillo, O., Melin, P., Montiel, O., Sepulveda, R., Pedrycz, W.: *Theoretical Advances and Applications of Fuzzy Logic and Soft Computing*. Springer, Tijuana (2007)
- [8]. Castillo, O., Neyoy, H., Soria, J., García, M., Valdez, F.: Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot. *Int. J. Adv. Robot. Syst.* (2013). <https://doi.org/10.5772/54883>
- [9]. Cázares-Castro, N.R., Aguilar, L.T., Castillo, O.: Designing type-1 and type-2 fuzzy logic controllers via fuzzy Lyapunov synthesis for nonsmooth mechanical systems. *Eng. Appl. AI* 25(5), 971–979 (2012)
- [10]. Goldberg, D.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston (1987)
- [11]. González, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput* 47, 631–643 (2016)
- [12]. González, C.I., Melin, P., Castro, J.R., Mendoza, O., Castillo, O.: An improved sobel edge detection method based on generalized type-2 fuzzy logic. *Soft Comput.* 20(2), 773–784 (2016)
- [13]. Gupta, P., Cambini, R., Appadoo, S.S.: Recent advances in optimization theory and applications (2018)
- [14]. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.* 39, 459–471 (2007)
- [15]. Kennedy, J., Eberhart, R.: Particle swarm optimization, pp. 1942–1948 (1995)
- [16]. Martinez, R., Castillo, O., Aguilar, L.T.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. *Inf. Sci* 179(13), 2158–2174 (2009)

- [17]. Melin, O.P.: Intelligent systems with interval type-2 fuzzy logic. *Int. J. Innov. Comput. Inf. Control* 4(4), 771–783 (2008)
- [18]. Melin, P., Amezcua, J., Valdez, F., Castillo, O.: A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias. *Inf. Sci* 279, 483–497 (2014) Design of Optimal Fuzzy Controllers for Autonomous Mobile Robots 295
- [19]. Melin, P., Castillo, O.: *Modelling, Simulation and Control of Non-linear Dynamical Systems: An Intelligent Approach Using Soft Computing and Fractal Theory*. CRC Press, Boca Raton (2001)
- [20]. Melin, P., Castillo, O.: Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory. *Appl. Soft Comput.* 3(4), 353–362 (2003)
- [21]. Melin, P., Sánchez, D., Castillo, O.: Genetic optimization of modular neural networks with fuzzy response integration for human recognition. *Inf. Sci* 197, 1–19 (2012)
- [22]. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* 69, 46–61 (2014)
- [23]. Muro, C., Escobedo, R., Spector, L., Coppinger, R.: Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behav. Process.* 88, 192–197 (2011)
- [24]. Olivas, F., Valdez, F., Castillo, O., Gonzales, C., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput.* 53, 74–87 (2016)
- [25]. Ramírez, C.L., Castillo, O., Melin, P., Díaz, A.R.: Simulation of the bird age structured population growth based on an interval type-2 fuzzy cellular structure. *Inf. Sci* 181(3), 519–535 (2011)
- [26]. Rodriguez, L., Castillo, O., Soria, J.: A Study of parameters of the grey wolf optimizer algorithm for dynamic adaptation with fuzzy logic. Springer, Tijuana (2017)
- [27]. Sanchez, M., Castillo, O., Castro, J.: Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. *Expert Syst. Appl.* 42, 5904–5914 (2015)
- [28]. Socha, M.D.K.: An introduction to ant colony optimization. In: *Handbook of Meta-heuristics*, vol. 26. IRIDIA, Brussels (2006). ISSN 1781-3794
- [29]. Sombra, A., Valdez, F., Melin, P., Castillo, O.: A new gravitational search algorithm using fuzzy logic to parameter adaptation. Cancun, México (2013)
- [30]. Soto, C., Valdez, F., Castillo, O.: A review of dynamic parameter adaptation methods for the firefly algorithm. In: *Nature-Inspired Design of Hybrid Intelligent Systems* (2007)
- [31]. Valdez, F., Melin, P., Castillo, O.: Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making (2009)
- [32]. Yang, X.S.: *Firefly Algorithm, Lévy Flights and Global Optimization* BT - Research and Development in Intelligent Systems, vol. XXVI (2010)
- [33]. Yang, X.S., Deb, S.: Cuckoo Search via Lévy Flights (2009)
- [34]. Yang, X.S., Karamanoglu, M.: Swarm intelligence and bio-inspired computation: an overview. In: *Swarm Intelligence and Bio-Inspired Computation*, pp. 3–23 (2013)
- [35]. Yang, X.S., Karamanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng. Optim.* 46(9), 1222–1237 (2014)

- [36]. Yang, X.S.: Bat: algorithm: literature review and applications. *Int. J. Bio-Inspired Comput.* 5(3), 141–149 (2013)
- [37]. Zadeh, L.: Fuzzy sets. *Inf. Control* 8, 338–353 (1965)
- [38]. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning—I. *Inform. Sci* 8, 199–249 (1975)
- [39]. Zadeh, L.: Fuzzy logic. *IEEE Comput. Mag.* 1, 83–93 (1988)