

Phát triển mô hình trí tuệ nhân tạo mới để tối ưu kế hoạch điều phối xe vận chuyển bê tông thương phẩm, góp phần giảm thiểu ô nhiễm khí thải

Phạm Vũ Hồng Sơn^{1*}, Nguyễn Thị Nha Trang²

¹ Khoa kỹ thuật xây dựng, trường Đại học Bách khoa thành phố Hồ Chí Minh (HCMUT), Đại học quốc gia thành phố Hồ Chí Minh (VNU-HCM)

² Khoa Kinh tế và quản lý xây dựng, trường Đại học xây dựng Hà Nội (HUCE)

TỪ KHÓA

Trí tuệ nhân tạo
Thuật toán tối ưu
Thuật toán sói xám
Thuật toán chuẩn chuẩn
Điều phối xe bê tông thương phẩm

TÓM TẮT

Bài báo này đề xuất một mô hình tối ưu hóa kết hợp thuật toán chuẩn chuẩn (Dragonfly Algorithm DA) và thuật toán sói xám (Grey Wolf Optimizer GWO). Trong mô hình này, thuật toán DA được sử dụng để cải thiện khả năng tìm kiếm toàn cục của thuật toán GWO. Trái ngược với những thuật toán GWO lai được phát triển trước đây, trong mô hình đề xuất của bài báo này, quá trình tối ưu hóa được dẫn dắt đồng thời bởi GWO và DA. Đầu tiên, quần thể các cá nhân tìm kiếm được chia làm hai nhóm. Mỗi nhóm được dẫn dắt bởi một thuật toán riêng rẽ. Sau đó, các nhóm con này được kết hợp lại thành một nhóm vào cuối mỗi lần lặp. Để xác minh chất lượng giải pháp của mô hình đề xuất, bài báo đã sử dụng một bài toán cụ thể về điều phối lịch trình xe tải vận chuyển bê tông thương phẩm. Kết quả của thuật toán được so sánh với hai thuật toán: thuật toán tối bầy đàn PSO và thuật toán tối ưu kiến sư tử ALO. Kết quả chỉ ra rằng mô hình lai được đề xuất vượt trội hơn PSO và ALO về chất lượng giải pháp, độ ổn định và khả năng tìm kiếm giải pháp tối ưu toàn cục.

KEYWORDS

Optimization
Swarm intelligence (SI)
Hybrid algorithm
Dragonfly algorithm (DA)
Grey wolf optimization (GWO) algorithm
RMC dispatch schedule

ABSTRACT

In this paper, a hybrid optimization model of Dragonfly algorithm and Grey Wolf Optimization is proposed. In this model, the dragonfly algorithm (DA) is combined with the grey wolf optimizer (GWO) to improve the global exploration ability of the GWO algorithm. In contrast to the hybrid GWO developed in the literature, in the proposed model, the optimization process is guided by the GWO and DA. The search process begins with the parallel and separate processing of two subgroups of a population with the GWO and DA; these subgroups are then combined into one group at the end of an iteration. To verify the solution quality of the proposed model, its performance was compared with that of the Particle swarm Optimization (PSO) and Ant Lion Optimization (ALO) by using a ready-mix concrete (RMC) dispatch schedule case study. The result indicates that the proposed hybrid model is superior to the PSO and ALO in terms of solution quality, stability, and capacity to discover the global optimum.

1. Giới thiệu

Tối ưu hóa là quá trình làm cho thứ gì đó tốt hơn, bao gồm việc thử nhiều lần các biến thể và sử dụng thông tin thu được để cải thiện nó ngày một tốt hơn. Các thuật toán tối ưu hứa hẹn sẽ giải quyết vấn đề này với rất nhiều ưu điểm. Trong các bài toán có sử dụng thuật toán tối ưu, đầu vào được đưa ra, hàm mục tiêu được xác định, các ràng buộc được thiết lập và quá trình tìm kiếm sẽ được thực hiện bên trong một không gian phức tạp. Quá trình tìm kiếm này hoàn toàn tự động, tiết kiệm thời gian và yêu cầu sự tham gia tối thiểu của con người.

Thuật toán metaheuristic là phương pháp tìm kiếm tối ưu mới. Hai loại chính của thuật toán metaheuristic bao gồm: thuật toán dựa trên cá nhân và thuật toán dựa trên tập hợp. Sự khác biệt giữa hai

loại thuật toán này liên quan đến số lượng tác nhân tìm kiếm. Trong các thuật toán metaheuristic dựa trên cá nhân, chỉ có một tác nhân tìm kiếm được tạo ra làm giải pháp ứng cử viên và tác nhân này được cải thiện sau các vòng lặp. Các thuật toán này dễ dàng rơi vào tối ưu cục bộ. Tuy nhiên, trong thuật toán metaheuristic dựa trên tập hợp, một tập hợp các giải pháp ngẫu nhiên (gọi là quần thể) được tạo khi bắt đầu quá trình tối ưu hóa và các giải pháp này được nâng cấp qua các lần lặp lại. Quá trình này làm giảm khả năng thuật toán rơi vào tối ưu cục bộ và tăng khả năng tìm thấy giá trị tối ưu toàn cục. Hạn chế của các thuật toán metaheuristic dựa trên tổng thể là tốc độ hội tụ thấp và các kết quả đầu ra có nhiều khả năng là khác nhau trong mỗi lần chạy.

Trí thông minh bầy đàn (SI) là một nhánh của thuật toán metaheuristic dựa trên tập hợp, được sử dụng để giải quyết các vấn

*Liên hệ tác giả: pvhson@hcmut.edu.vn

Nhận ngày 30/02/2022, sửa xong ngày 31/03/2022, chấp nhận đăng 08/09/2022

Link DOI: <https://doi.org/10.54772/jomc.05.2022.406>

để tối ưu hóa. Các thuật toán SI dựa trên hành vi của bầy đàn, cộng đồng hoặc bất kỳ hệ thống nào trong tự nhiên. Bầy bao gồm bầy thú, bầy chim, bầy côn trùng, đàn cá, tất cả đều có nhiều thành viên. Các thuật toán SI mô phỏng hành vi của các thành viên trong bầy và sự tương tác và quan hệ giữa các thành viên này khi tiếp cận nguồn thức ăn. Các nhà nghiên cứu đã phát triển nhiều thuật toán SI, như là thuật toán bầy đàn (Kennedy and Eberhart 1995), thuật toán đom đóm (Yang 2009), thuật toán sói xám (Mirjalili, Mirjalili et al. 2014), Thuật toán đàn ong nhân tạo (Basturk 2006), thuật toán kiến sư tử (Mirjalili 2015), thuật toán tối ưu hóa loài bướm đêm (Mirjalili 2015), thuật toán cá voi (Mirjalili and Lewis 2016), thuật toán chuồn chuồn (Mirjalili 2016)....

Theo định lý No Free Lunch (Wolpert and Macready 1997) không có thuật toán nào tốt nhất cho tất cả các bài toán. Mỗi thuật toán đều có ưu điểm và nhược điểm của nó. Tùy theo mục tiêu cần đạt của bài toán mà sử dụng thuật toán phù hợp. Thuật toán chất lượng là thuật toán cân bằng tốt giữa hai giai đoạn thăm dò và khai thác, giữa kết quả tìm được và thời gian chạy chương trình. Các nhà nghiên cứu đều đang cố gắng tìm cách làm cho một thuật toán tốt hơn để cung cấp nhiều sự lựa chọn hơn cho các chuyên gia và người dùng trong các lĩnh vực khác nhau.

Trong nghiên cứu này, một thuật toán tối ưu hóa mới được đề xuất, là sự kết hợp của hai thuật toán sói xám và chuồn chuồn. Các phần sau sẽ trình bày chi tiết hơn về thuật toán này và hiệu quả của nó trong việc giải quyết bài toán tối ưu lịch trình phân phối bê tông thương phẩm. Nghiên cứu sẽ tập trung vào việc ứng dụng thuật toán này để tìm ra được chuỗi cung ứng cho vấn đề phân phối bê tông thương phẩm bằng xe chuyên dụng sao cho “tổng thời gian chờ đợi của xe bê tông tại công trường và thời gian công trường chờ xe bê tông tới là thấp nhất”.

2. Nghiên cứu tổng quan

Các thuật toán tối ưu hóa khác nhau có những hạn chế khác nhau; do đó, hai hoặc nhiều thuật toán tối ưu ngẫu nhiên thường được kết hợp để tích hợp các ưu điểm và khắc phục nhược điểm của chúng. Đặc biệt, một số phương pháp tiếp cận kết hợp đã được xây dựng mà trong đó thuật toán GWO được lai ghép với các thuật toán khác.

Từ khi ra đời, thuật toán GWO đã thu hút sự quan tâm ngày càng tăng từ các nhà nghiên cứu, những người đã cố gắng sửa đổi thuật toán GWO để phù hợp với không gian tìm kiếm của các miền phức tạp. Vào năm 2017, Singh và Singh (Singh and Singh 2017) đã đề xuất một phương pháp lai liên quan đến việc kết hợp thuật toán GWO và thuật toán sin cos (SCA). Singh và Singh chỉ thay đổi phương trình cập nhật vị trí của con sói alpha trong thuật toán GWO bằng cách thay thế nó bằng phương trình cập nhật vị trí của SCA. Bằng cách sử dụng một quy trình tương tự, Pan (Pan, Dao et al. 2017) kết hợp thuật toán GWO với thuật toán thụ phấn hoa (FPA). Chuyển động của các phần tử trong thuật toán GWO đã được cải thiện bằng cách sử dụng cập nhật vị trí của thuật toán FPA.

Chopra (Chopra, Kumar et al. 2016) và Kamboj (Kamboj 2016) là những người đầu tiên kết hợp các thuật toán GWO và PSO. Họ lần lượt sử dụng các thuật toán này ở mỗi lần lặp lại. Các tác giả đã đề cập ở trên cho rằng tập hợp giải pháp được cải thiện bằng cách sử dụng các điểm mạnh của từng thuật toán một cách riêng biệt ở mỗi lần lặp lại. Tuy nhiên, các phương pháp của Chopra và Kamboj khác nhau ở một khía cạnh chính. Trong phương pháp Chopra (Chopra, Kumar et al. 2016), toàn bộ quần thể thu được từ một thuật toán trong lần lặp này được dùng làm quần thể của thuật toán kia trong lần lặp tiếp theo. Ngược lại, trong phương pháp Kamboj (Kamboj 2016), chỉ cá thể tốt nhất thu được từ một thuật toán được sử dụng để cập nhật quần thể của thuật toán khác trong lần lặp tiếp theo. Mặc dù các phương pháp nói trên có chất lượng cao nhưng tốc độ hội tụ của chúng thấp do việc thực hiện tuần tự từng thuật toán làm kéo dài thời gian.

Singh và Singh (Singh and Singh 2017) cũng đề xuất sự kết hợp giữa các thuật toán GWO và PSO. Ngược lại với các mô hình của Chopra (Chopra, Kumar et al. 2016) và Kamboj (Kamboj 2016), các thuật toán GWO và PSO không chạy lần lượt mà thay vào đó là chạy song song. Singh và Singh đã sửa đổi phương trình GWO bằng cách thêm một hằng số quán tính vào để điều khiển việc tìm kiếm và khai thác của các tác nhân trong không gian tìm kiếm. Các phương trình vận tốc và cập nhật vị trí của thuật toán PSO cũng được sửa đổi để xem xét ba cá thể tốt nhất của thuật toán GWO.

Senel (Senel, Gökçe et al. 2019) đề xuất sự kết hợp của thuật toán GWO và PSO, trong đó thuật toán GWO được sử dụng để hỗ trợ thuật toán PSO. Đầu tiên, một phần quần thể của thuật toán PSO được lựa chọn với xác suất thấp và được cải thiện bởi thuật toán GWO qua một số lần lặp. Sau đó, tập hợp này được thay thế bằng giá trị trung bình của các giá trị thu được bởi những con sói tốt nhất của thuật toán GWO.

Các thuật toán như thuật toán tiến hóa khác biệt DE cũng đã được kết hợp với thuật toán GWO algorithm (Zhu (Zhu, Xu et al. 2015) and Jitkongchuen (Jitkongchuen 2015)).

Các thuật toán lai nói trên hoạt động tốt hơn các thuật toán riêng lẻ. Tuy nhiên, các nỗ lực vẫn đang được tiến hành để cải thiện khả năng hội tụ và tìm kiếm toàn cục của thuật toán GWO. Hơn nữa, định lý NFL (Wolpert and Macready 1997) tạo động lực thúc đẩy nghiên cứu này để xuất thuật toán lai GWO mới.

3. Phát triển thuật toán tối ưu mới

3.1. Thuật toán sói xám

Seyedali (Mirjalili, Mirjalili et al. 2014) giới thiệu GWO vào năm 2014. Thuật toán này lập mô hình toán học hành vi của sói xám khi chúng săn mồi theo bầy. Một bầy sói bao gồm bốn loại: alpha (lãnh đạo bầy đưa ra quyết định săn bắn), beta, delta và omega, theo thứ tự ở cấp độ giảm dần của hệ thống phân cấp xã hội của bầy sói. Alpha, beta, delta lần lượt đại diện cho các giải pháp tốt nhất, tốt thứ

hai và tốt thứ ba. Những con sói omega (ω), đại diện cho giải pháp còn lại của quần thể.

Trong thuật toán GWO, quá trình bầy sói đuổi theo và bao vây con mồi được mô phỏng như các tác nhân di chuyển để tìm ra giải pháp tối ưu. Sau mỗi lần lặp lại, những con sói (các tác nhân) cập nhật vị trí của chúng theo vị trí của con mồi (giải pháp tốt nhất thu được từ lần lặp cuối cùng) như được mô tả trong các phương trình sau:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Trong đó $\vec{X}(t)$ là vị trí của con sói ở vòng lặp t , $\vec{X}_p(t)$ là véc tơ vị trí của con mồi, \vec{A} và \vec{C} là các véc tơ hệ số. Những véc tơ này được tính như sau:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

Trong đó \vec{a} giảm tuyến tính từ 2 về 0 qua các vòng lặp, r_1 và r_2 là các hệ số ngẫu nhiên trong khoảng [0, 1].

Trong thuật toán GWO, vị trí của con mồi (hay là giá trị tối ưu nhất) không được biết trước. Các con sói alpha, beta, và delta được giả định là biết về vị trí con mồi, vì thế các con sói trong đàn sẽ đi theo ba con sói này và cập nhật vị trí mới của mình theo các phương trình sau:

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (5)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (6)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (7)$$

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (8)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (9)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (10)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (11)$$

Trong đó $X(t+1)$ thể hiện vị trí của con sói ở vòng lặp $(t+1)$.

Để đạt được sự cân bằng tốt hơn giữa các giai đoạn tìm kiếm và khai thác, \vec{A} và \vec{C} được sử dụng để kiểm soát quá trình săn mồi. Quá trình tìm kiếm được thúc đẩy bởi \vec{A} . Khi giá trị của \vec{A} nằm trong $[-1, 1]$, sói có nghĩa vụ phải tách khỏi con mồi. Khi \vec{A} có giá trị ngẫu nhiên trong $[-1, 1]$, quá trình khai thác bắt đầu. Trong pha này, sói xám đến gần con mồi và tấn công nó. Do đó, các tác nhân tìm kiếm hội tụ về giải pháp tốt nhất trong số các giải pháp alpha, beta và delta, điều này ngăn chúng bị mắc kẹt vào cực tiểu cục bộ.

Tham số khác đảm bảo cho việc tìm kiếm là \vec{C} . Véc tơ này nhận các giá trị ngẫu nhiên trong [0, 2] trong quá trình tối ưu hóa. Do đó, nó có thể gây ra những thay đổi đột ngột về vị trí của các tác nhân tìm kiếm để khám phá ra khu vực đầy hứa hẹn của không gian tìm kiếm. Khi $C > 1$, các tác nhân tìm kiếm sẽ hội tụ về phía con mồi, và khi $C < 1$, các tác nhân tìm kiếm chuyển hướng khỏi con mồi.

3.2. Thuật toán chuồn chuồn

Mirjalili (Mirjalili 2016) đã giới thiệu DA vào năm 2016. Thuật toán này được lấy cảm hứng từ hành vi của chuồn chuồn trong tự nhiên. Chuồn chuồn là loài côn trùng nhỏ sống thành bầy đàn để săn

mồi và di cư, và cách sống bầy đàn của chúng khác nhau tùy theo từng mục đích. Để săn mồi, chuồn chuồn hình thành những đàn nhỏ bay qua các khu vực khác nhau để đuổi theo con mồi. Hành vi này được gọi là bầy đàn tĩnh (static swarming). Để di cư, chuồn chuồn tạo thành bầy lớn và di chuyển theo một hướng. Hành vi này được gọi là bầy đàn động (dynamic swarming). Các hành vi bầy đàn nói trên có các đặc điểm tương tự như các giai đoạn thăm dò và khai thác của quá trình tối ưu hóa metaheuristic.

Năm yếu tố được sử dụng để đại diện cho hai kiểu bầy đàn của chuồn chuồn: phân vùng (partition), sắp xếp (arrangement), gắn kết (cohesion), thu hút đối với nguồn thức ăn (attraction toward a food source) và chuyển hướng khỏi kẻ thù (diversion away from an adversary). Các yếu tố này được xác định như sau.

Phân vùng được tính như sau:

$$S_i = - \sum_{j=1}^N X - X_j \quad (12)$$

Trong đó X là vị trí hiện tại của chuồn chuồn, X_j là vị trí của chuồn chuồn thứ j gần đó và N là số chuồn chuồn gần đó.

Sự sắp xếp được tính như sau:

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (13)$$

Với V_j thể hiện vận tốc của chuồn chuồn thứ j gần đó.

Gắn kết được tính bằng công thức sau:

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (14)$$

Thu hút đối với nguồn thức ăn được tính như sau:

$$F_i = X^+ - X \quad (15)$$

Với X và X^+ lần lượt thể hiện vị trí của cá thể chuồn chuồn hiện tại và nguồn thức ăn.

Chuyển hướng khỏi kẻ thù được tính theo công thức sau:

$$E_i = X^- - X \quad (16)$$

Với X và X^- lần lượt thể hiện vị trí của cá thể chuồn chuồn hiện tại và kẻ thù.

Để cập nhật vị trí của chuồn chuồn trong không gian tìm kiếm, vectơ bước nhảy có thể được tính như sau:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + \omega \Delta X_t \quad (17)$$

Trong đó s là trọng số phân vùng, S_i là phân vùng của chuồn chuồn thứ i , a là trọng số liên kết, A_i là liên kết của chuồn chuồn thứ i , c là trọng số gắn kết, C_i là độ nhất quán của chuồn chuồn thứ i , f là hệ số thức ăn, F_i thể hiện nguồn thức ăn của chuồn chuồn thứ i , e là hệ số kẻ thù, E_i là vị trí kẻ thù của chuồn chuồn thứ i , ω là trọng số quán tính, và t là vòng lặp hiện tại.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iteration_{max}} \cdot Iteration_t \quad (18)$$

Trong công thức (18), ω_{max} được đặt là 0,9; ω_{min} được đặt là 0,4; và $Iteration_{max}$ số lượng vòng lặp tối đa.

Véc tơ vị trí được tính như sau:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (19)$$

Bằng cách điều chỉnh các thành phần s , a , c , f , và e , các hành vi khám phá và khai thác khác nhau có thể đạt được trong quá trình tối ưu hóa.

Khi không có giải pháp lân cận nào tồn tại, vị trí của chuẩn chuẩn nhân tạo được cập nhật bằng bước nhảy ngẫu nhiên (Levy flight) như sau:

$$X_{t+1} = X_t + Levy(d) \times X_t \quad (20)$$

Khu vực lân cận dần dần được mở rộng, và bầy đàn cuối cùng chỉ tạo thành một nhóm trong giai đoạn cuối cùng của quá trình tối ưu hóa. Cuối cùng, giải pháp tốt nhất và kém nhất mà quần thể thu được lần lượt là vị trí của nguồn thức ăn và kẻ thù.

3.3. Thuật toán lai ghép

Thuật toán GWO đã được sử dụng trong một số lĩnh vực nghiên cứu. Ưu điểm chính của thuật toán này sự đơn giản trong công thức tính toán, ít tham số điều khiển [18]. Tuy nhiên, tương tự như các thuật toán metaheuristic khác, thuật toán GWO có một số hạn chế.

Hạn chế chính của thuật toán GWO là khả năng xử lý không gian tìm kiếm multimodal thấp (Faris, Aljarah et al. 2018). Hạn chế này xảy ra bởi vì các con sói alpha, beta và gamma có xu hướng hội tụ về cùng một giải pháp. Thuật toán lai ghép được phát triển trong nghiên cứu này khắc phục nhược điểm trên của thuật toán GWO bằng cách chia quần thể tác nhân tìm kiếm thành hai nhóm và cung cấp cho các tác nhân cơ hội khám phá các phần khác của không gian tìm kiếm thay vì tiến về phía ba con sói tốt nhất. DA được kết hợp với GWO trong nghiên cứu này vì DA có khả năng tìm kiếm vượt trội, cho phép nó tìm kiếm ở những khu vực đầy hứa hẹn trong không gian tìm kiếm.

Thuật toán lai của nghiên cứu này được mô tả như sau. Đầu tiên, các tác nhân được xếp hạng theo giá trị hàm mục tiêu của nó, và quần thể sau đó được chia thành hai nhóm. Các thành viên của nhóm 1 được chọn từ các tác nhân có xếp hạng giá trị hàm mục tiêu từ 1 đến N/2 và các tác nhân còn lại của quần thể sẽ lập nên nhóm 2. Sau đó, thuật toán GWO được chạy với nhóm 1 và đồng thời DA được chạy với nhóm 2. Hai thuật toán không chạy nối tiếp nhau; thay vào đó, chúng chạy song song với nhau. Cuối cùng, tất cả các tác nhân từ hai nhóm được kết hợp trở lại thành một nhóm khi kết thúc một lần lặp.

Do DA có khả năng tìm kiếm cao nên các tác nhân trong nhóm 2 tránh được rơi vào trạng thái tối ưu cục bộ; thay vào đó, DA hướng các tác nhân này tìm kiếm trong các khu vực đầy hứa hẹn của không gian tìm kiếm. Trong phương pháp kết hợp, hai thuật toán không ảnh hưởng đến tính ổn định của nhau mà hỗ trợ nhau trong việc thăm dò và khai thác.

Các bước liên quan đến thuật toán DA – GWO được đề xuất như sau.

- Bước 1. Đặt các tham số và số lần lặp của thuật toán DA và GWO.
- Bước 2. Khởi tạo quần thể ban đầu.
- Bước 3. Tính toán các giá trị hàm mục tiêu cho quần thể ban đầu.
- Bước 4. Kiểm tra xem một cá thể có giá trị hàm mục tiêu xếp hạng từ 1 đến N/2. Nếu có, thêm cá thể này vào nhóm 1, cụ thể là quần thể sói, và nếu không, thêm cá thể vào nhóm 2, cụ thể là quần thể chuẩn chuẩn.
- Bước 5. Đối với nhóm 1:
 - Bước 5.1 Tính toán ra 3 giá trị tốt nhất (ba con sói tốt nhất).

Bước 5.2 Đặt các giá trị của a, A, và C.

Bước 5.3 Tính X1, X2 và X3 bằng cách sử dụng các công thức (5) - (7), tương ứng.

Bước 5.4 Sử dụng công thức (11) để tính vector vị trí.

Bước 6 Đối với nhóm 2:

Bước 6.1 Cập nhật vị trí của nguồn thức ăn và kẻ thù.

Bước 6.2 Cập nhật w, s, a, c, f, và e.

Bước 6.3 Tính S, A, C, F, và E bằng cách sử dụng các công thức (12) - (16), tương ứng.

Bước 6.4 Cập nhật bán kính lân cận nếu có ít nhất một con chuẩn chuẩn ở gần một con chuẩn chuẩn đang được xem xét.

Bước 6.5 Cập nhật véc tơ vận tốc bằng cách sử dụng công thức (17).

Bước 6.6 Cập nhật vector vị trí bằng cách sử dụng công thức (19).

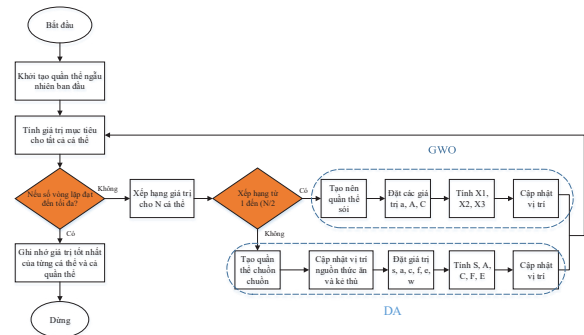
Nếu không thì:

Cập nhật vector vị trí bằng cách sử dụng công thức (20).

Bước 7. Tính các hàm mục tiêu của tất cả các cá thể trong cả hai nhóm và lưu giá trị tốt nhất.

Bước 8. Kiểm tra điều kiện dừng (số lần lặp). Nếu điều kiện đạt được, dừng quá trình; nếu điều kiện không đạt được, hãy quay lại bước 4.

Lưu đồ của thuật toán lai DA-GWO được thể hiện trong Hình 1.



Hình 1. Lưu đồ thuật toán lai.

4. Ứng dụng mô hình vào giải quyết bài toán điều phối xe vận chuyển bê tông thương phẩm

Bê tông trộn sẵn hay còn gọi là bê tông thương phẩm (BTTP) là một trong những loại vật liệu phổ biến nhất trong ngành xây dựng hiện nay. Các vật liệu thành phần của BTTP được cân đo và trộn sẵn tại nhà máy bằng thiết bị chuyên dụng tự động hoặc được đưa vào thùng xe tải vận chuyển bê tông và trộn trong quá trình lưu thông trên đường đến địa điểm xây dựng, đảm bảo các yêu cầu kỹ thuật của bê tông cung cấp cho công trình. BTTP mang lại sự tiện lợi cho khách hàng sử dụng do tính ưu việt của nó về tiết kiệm thời gian, giảm chi phí, độ bền cao, không yêu cầu kho dự trữ và xử lý vật liệu.

Tuy nhiên, BTTP lại không thể được sản xuất trước khối lượng lớn và lưu trữ sẵn trong kho do đặc tính khô nhanh của bê tông và do yêu cầu chất lượng bê tông ở các công trình là khác nhau. BTTP thường cần sử dụng trong vòng 2 tiếng sau khi sản xuất để đảm bảo chất lượng ổn định. Ngoài ra, BTTP cần được đổ liên tục tại công

trường để tránh mạch ngừng trong thi công. Vì vậy việc vận chuyển BTTP mà bị chậm trễ sẽ gây ra chất lượng bê tông không đảm bảo. Tuy nhiên có rất nhiều yếu tố có thể ảnh hưởng đến việc vận chuyển BTTP như là xe bị hỏng trên đường đi, độ chính xác của việc tính toán khoảng thời gian di chuyển, tình trạng giao thông, và giới hạn số lượng xe tải mà nhà máy sở hữu. Vì thế việc tối ưu quá trình sản xuất và lịch trình vận chuyển BTTP là một vấn đề hết sức cần thiết. Nghiên cứu này đề xuất một mô hình điều phối các xe tải bê tông trộn sẵn bằng cách sử dụng thuật toán tối ưu HMDG để quyết định một lịch trình tối ưu nhằm giảm thiểu thời gian chờ đợi của các xe tải tại công trình (TWC) và thời gian công trình phải chờ đợi xe tải (CWT).

4.1. Vấn đề điều phối lịch trình xe bê tông

Các thông số đầu ra (outputs): mục tiêu của bài toán điều động RMC là giảm thiểu tổng thời gian tối đa mà xe tải phải chờ tại công trường và giảm thiểu tổng thời gian tối thiểu mà công trường phải chờ xe tải (Feng, Cheng et al. 2004), (Feng and Wu 2006). Để làm được điều đó, bài báo phát triển một trình tự điều phối để cung cấp RMC đến các công trường xây dựng một cách hiệu quả. Ngoài ra, nghiên cứu đưa ra lịch trình chi tiết cho từng xe tải để tạo ra một sản phẩm có thể áp dụng ngay cho công trường mà không cần nhà phân phối.

Các thông số đầu vào (inputs): Các thông số này bao gồm thời gian bắt đầu đổ bê tông tại công trường (CST), thời gian đổ bê tông tại công trường (CD), thời gian di chuyển từ trạm trộn đến địa điểm xây dựng (TDG), thời gian di chuyển từ địa điểm xây dựng trở lại trạm (TDB), thời gian trộn tại trạm (MD), và số lượng xe tải thuộc sở hữu của trạm trộn.

Các biến số quyết định (decision variable): Trong nghiên cứu này, chúng ta cần tìm các trình tự điều động xe tải chờ bê tông để tối ưu hóa các mục tiêu của bài toán. “Trình tự điều động” được định nghĩa là trình tự điều động từng xe tải đến các địa điểm xây dựng khác nhau (Feng, Cheng et al. 2004). Trình tự này là biến quyết định duy nhất được coi là liên quan đến lịch trình kiểm soát. Ví dụ, định dạng dữ liệu được biểu diễn bằng vectơ [1.1.1.2.2.2.2], với các số 1 và 2 đại diện cho hai vị trí xây dựng tương ứng. Trong ví dụ này, ba xe tải đầu tiên sẽ được chuyển đến công trình 1 và bốn xe tải tiếp theo sẽ được đưa đến công trình 2.

Hạn chế (constraints): Thời gian chờ xe tải RMC đến công trường phải ít hơn thời gian đệm cho phép và thời gian chờ tại công trường không quá lâu để tránh lãng phí nhân lực. Ngoài ra, sức chứa của xe tải và số lượng xe tải của trạm là có hạn. Trên thực tế, mỗi trạm trộn bê tông sở hữu một đội xe tải hạn chế và mỗi xe tải có giới hạn tải cố định, thường là 6m³. Vì vậy, khi công trường đặt hàng 22 m³ bê tông thì trạm bê tông phải cử ít nhất 4 xe tải loại 6 m³ mới giao xong (trong đó có 3 xe tải chờ 6m³ và 1 xe tải chờ 4m³).

4.2. Quá trình mô phỏng

Bước 1: Xác định thời gian chờ đợi tối thiểu

Hai mục tiêu cần được tối ưu hóa.

$$\text{Minimize: } TWC = \sum_{ij} \max(0, WC_{ji}) \text{ nếu } WC_{ji} \geq 0 \quad (21)$$

$$\text{Minimize: } CWT = \sum_{ij} \min(0, WC_{ji}) \text{ nếu } WC_{ji} < 0 \quad (22)$$

Trong đó WC_{ji} là thời gian xe tải thứ i đợi ở công trường j (thể hiện một giá trị dương). WC_{ji} cũng là lúc công trường j phải chờ xe tải i (hiện giá một giá trị âm).

Bước 2: Khai báo các thông số

Người dùng phải nhập các thông số cụ thể cho mô hình điều khiển, bao gồm số xe tải, SCT, CD, TDG, TDB, MD, DJ.

Bước 3: Xác định thời gian khởi hành lý tưởng cho mỗi xe chờ bê tông, xác định bằng công thức (23-25).

$$FDT = \min_{j=1}^m (SCT_j - TDG_j) \quad (23)$$

$$IDT_i = FDT + (i - 1)MD_i \quad i = 1 \sim N \quad (24)$$

$$N = \sum_{j=1}^m k_j \quad (25)$$

Trong đó FDT là thời gian khởi hành lý tưởng cho xe tải đầu tiên của trạm; SCT_j là thời điểm bắt đầu đổ bê tông tại công trường; TDG_j là thời gian di chuyển từ trạm đến công trường j ; IDT_i là thời gian khởi hành lý tưởng cho xe i ; MD_i là thời gian trộn bê tông tại trạm; m là số lượng công trình xây dựng cần xe chờ bê tông; i là thứ tự của các xe tải; k_j là số lượng lượt xe cần thiết để chờ bê tông đến công trường j ; và N là tổng số xe tải cần thiết của tất cả các công trường.

Bước 4: Mô phỏng quá trình phân phối RMC. Thời gian khởi hành lý tưởng của xe đầu tiên c có thể được xác định theo công thức (23) - (25).

Tuy nhiên, không có xe tải nào có thể đã quay trở lại sau khi tất cả các xe tải rời đi. Do đó, thời gian phân bổ cho các xe tải còn lại được tính bằng quy trình mô phỏng được trình bày trong Công thức (26), (27) dựa trên thứ tự điều động do mô hình HMDG tạo ra.

$$STD_i = IDT_i \quad \text{if } i \leq c \quad (26)$$

$$STD_i = \min_i (TBB_i + MD_i) \quad \text{if } c < i \leq N \quad (27)$$

Đối với lượt xe tải đầu tiên trong tổng số xe (c), STD được tính theo công thức (26). Đối với các lượt xe sau, STD được tính bằng (27). TBB_i là thời điểm các xe tải của lượt sau đã về trạm nhưng chưa được gửi đi. J là chỉ số của địa điểm xây dựng được ký hiệu $j = 1 \rightarrow m$, và c là số xe tải bê tông thuộc sở hữu của trạm. Thời gian xe thứ i đến công trường j là TAC_{ji} được tính như sau:

$$TAC_{ji} = SDT_i + TDG_{ji} \quad (28)$$

PTF_{ji} là thời điểm công trường bắt đầu đổ bê tông; LT_j là thời gian xe tải thứ i rời khỏi công trường; TBB_i là thời gian xe chờ bê tông đầu tiên quay trở lại trạm theo công thức 34.

$$PTF_{ji} = SCT_j, \text{ if } i \leq c \quad (29)$$

$$PTF_{ji} = LT_{j(k-1)}, \text{ if } c < i \leq c \quad (30)$$

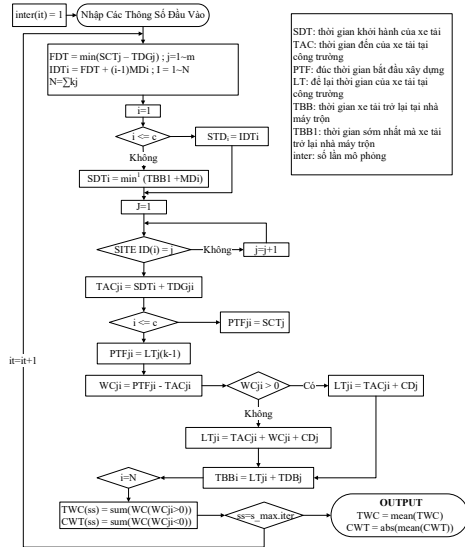
$$WC_{ji} = PTF_{ji} - TAC_{ji} \quad (31)$$

$$LT_{ji} = TAC_{ji} + WC_{ji} + CD_j \text{ if } WC_{ji} \leq 0 \quad (32)$$

$$LT_{ji} = TAC_{ji} + CD_j, \text{ if } WC_{ji} > 0 \quad (33)$$

$$TBB_i = LT_{ji} + TDB_j \quad (34)$$

Hình 2 thể hiện quy trình điều phối xe bê tông từ trạm trộn đến các công trình.



Hình 2. Mô phỏng phân phối bê tông từ trạm đến các công trình.

4.3. Áp dụng vào trường hợp cụ thể

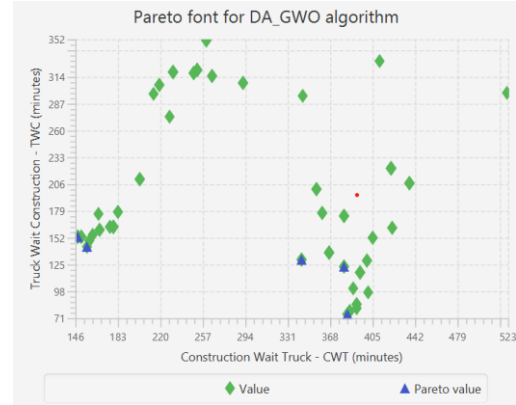
Các thông tin chi tiết của bài toán điều phối xe bê tông như trong Bảng 1.

Bảng 1. Các thông số đầu vào.

Công trình	Thời gian bắt đầu làm việc của công trường (SCT)	Khối lượng bê tông đặt mua	Chi tiết loại bê tông công trường đặt mua (PT)	Khoảng cách từ trạm trộn đến công trường (DJ-Km)
1	8:00	16	Sàn	10
2	8:15	22	Dầm	8
3	8:30	30	Cột	15
4	8:00	20	Sàn	10
5	8:00	25	Dầm	9
6	8:30	32	Cột	8
7	8:00	14	Sàn	10
8	9:00	18	Dầm	6

Số lượng xe trạm sở hữu: 10 xe
 Khối lượng bê tông mỗi xe có thể chở: 6 m³
 Vận tốc đi: 40 km/h;
 Vận tốc về: 50 km/h
 Thời gian trộn: 3 phút/m³
 Tốc độ đổ bê tông các cấu kiện (CD)
 Sàn: 4 phút/m³
 Dầm: 5 phút/m³
 Cột: 6 phút/m³

Sử dụng máy tính xách tay cá nhân với bộ xử lý Intel Core i5 và chạy mỗi thuật toán 30 lần với 100 tác nhân tìm kiếm trong 100 lần lặp, chúng tôi đã thu được mặt phẳng tối ưu Pareto như trong Hình 3.



Hình 3. Biểu đồ Pareto thể hiện các giá trị tối ưu tìm được bằng thuật toán lai DA-GWO.

Thời gian hoàn thành toàn bộ quá trình tìm kiếm là 459 giây. Thuật toán mang lại năm giá trị Pareto và hai giá trị tối ưu cho hai mục tiêu tìm kiếm. Kết quả tìm kiếm chi tiết được cung cấp trong bảng sau.

Bảng 2. Các giải pháp Pareto và chuỗi phân phối.

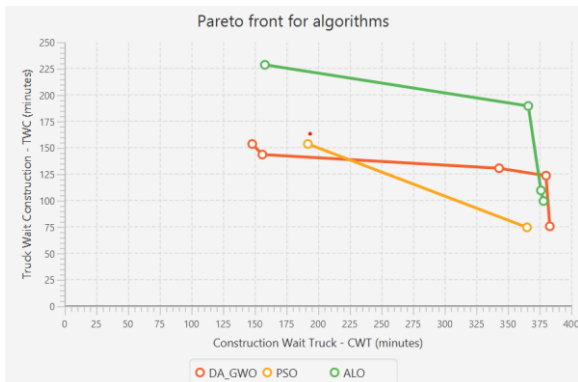
Giá trị Pareto	Các giá trị hàm mục tiêu tương ứng	Chuỗi phân phối
R1	= > TWC: 143.0 = > CWT: 156.0	[1,7,4,5,1,2,7,3,4,6,1,7,4,6,3,2,5,4,8,2,6,3,5,8,6,2,3,6,8,5,3,6,5]
R2	= > TWC: 153.0 = > CWT: 148.0	[7,4,1,5,2,6,7,1,3,4,7,1,4,2,6,3,8,4,5,6,2,3,8,5,6,3,2,8,5,6,3,5,6]
R3	= > TWC: 130.0 = > CWT: 343.0	[1,7,4,5,3,5,6,7,2,4,2,3,6,6,5,4,3,2,7,8,1,6,3,8,5,1,4,2,6,3,5,8,6]
R4	= > TWC: 123.0 = > CWT: 380.0	[1,7,4,5,3,4,5,6,2,7,3,2,6,4,3,6,8,5,3,2,5,7,6,1,4,1,8,6,3,5,2,8,6]
R5	= > TWC: 75.0 = > CWT: 383.0	[1,5,4,7,3,4,5,2,7,6,3,2,6,4,5,3,6,8,2,1,3,5,6,8,4,7,2,6,1,5,8,3,6]

Trong đó R1, R2, R3, R4, R5 lần lượt là các giá trị Pareto trong hình 3 theo chiều từ trái sang phải.

Để kiểm tra tính hiệu quả của thuật toán lai, chúng tôi so sánh kết quả của thuật toán với các thuật toán PSO và ALO. Kết quả nêu trong Bảng 3 và biểu đồ Pareto ở hình 4 chỉ ra rằng thời gian tối ưu thu được với thuật toán lai DA-GWO tốt hơn nhiều so với thời gian thu được từ PSO và ALO ở hàm mục tiêu CWT, với kết quả nhỏ hơn lần lượt là 44 và 10 phút. Ở hàm mục tiêu TWC, thuật toán lai cho kết quả tốt hơn thuật toán ALO khi giảm thời gian được 24 phút và cho kết quả bằng thuật toán PSO khi cùng cho giá trị tối ưu là 75 phút. Ngoài ra thuật toán lai còn mang lại nhiều giá trị Pareto hơn so với hai thuật toán còn lại. Điều này mang lại cho người ra quyết định nhiều sự lựa chọn hơn theo các tiêu chí và mức độ ưu tiên khác nhau của các hàm mục tiêu. Những kết quả này chỉ ra rằng thuật toán lai cung cấp các giải pháp tốt hơn so với hai thuật toán còn lại.

Bảng 3. Kết quả của các thuật toán khác nhau.

Algorithm	CWT (minutes)	TWC (minutes)	Số lượng giá trị Pareto tìm được
DA-GWO	148	75	5
PSO	192	75	2
ALO	158	99	4



Hình 4. Biểu đồ Pareto thể hiện các giá trị tối ưu tìm được bằng thuật toán khác nhau.

Việc sử dụng thuật toán tối ưu đã làm giảm thời gian chờ đợi của xe tại công trường (TWC) từ 411 phút (kết quả kém nhất trong tất cả các vòng lặp) xuống 75 phút (giá trị tốt nhất trong tất cả các lần lặp). Tổng thời gian tiết kiệm được là 336 phút. Sử dụng quá trình tối ưu hóa sẽ giảm được thời gian vận hành của xe không cần thiết, từ đó giảm được lượng khí thải phát ra, góp phần bảo vệ môi trường. Theo (Ntziachristos, Giachaskiel et al. 2003), những xe sử dụng động cơ diesel như xe vận chuyển BTTP có mức phát thải trung bình là: CO: 0.64g/km; HC + NOx: 0,56g/km, NOx: 0,5g/km, PM: 0,05g/km. Trong bài báo này, giả thiết xe vận chuyển BTTP chạy khi có tải là 40km/giờ, thời gian tiết kiệm được 336 phút tương đương với giảm được 224km. Mức phát thải giảm được như trong bảng sau:

Bảng 4. Mức giảm khí thải.

Khí thải	Mức phát thải (g/km)	Mức giảm phát thải (g)
CO	0,64	143,36
HC + NOx	0,56	125,44
NOx	0,5	112
PM	0,05	11,2

5. Kết quả

Trong nghiên cứu này, một thuật toán metaheuristic lai được phát triển bằng cách kết hợp thuật toán GWO và DA. Thuật toán lai này sử dụng khả năng tìm kiếm của DA để hỗ trợ thuật toán GWO không bị rơi vào tối ưu cục bộ. Trong thuật toán lai, các tác nhân tìm kiếm được chia thành hai nhóm có kích thước bằng nhau theo thứ hạng về giá trị hàm mục tiêu của chúng. Nhóm 1 bao gồm các tác

nhân có xếp hạng từ 1 đến N/2 và dẫn dắt bởi thuật toán GWO. Nhóm 2 bao gồm các tác nhân còn lại và được dẫn dắt bởi DA. Một bài toán thực tế về điều phối xe bê tông thương phẩm đã được sử dụng để kiểm tra hiệu quả của thuật toán lai. Kết quả của thuật toán lai đã được so sánh với kết quả của các thuật toán khác, bao gồm PSO và ALO. Kết quả chỉ ra rằng thuật toán HDGM thực hiện thăm dò và khai thác đáng tin cậy hơn so với thuật toán PSO và ALO.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Bách Khoa Thành phố Hồ Chí Minh- Đại học Quốc gia Thành phố Hồ Chí Minh thông qua đề tài mã số C2021-20-34. Các tác giả chân thành cảm ơn Trường Đại học Bách Khoa Thành phố Hồ Chí Minh- Đại học Quốc gia Thành phố Hồ Chí Minh đã hỗ trợ về thời gian và cơ sở vật chất cho nghiên cứu này.

Tài liệu tham khảo

- [1]. Basturk, B. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2006.
- [2]. Chopra, N., et al. (2016). "Hybrid GWO-PSO algorithm for solving convex economic load dispatch problem." Int J Res Adv Technol 4(6): 37-41.
- [3]. Faris, H., et al. (2018). "Grey wolf optimizer: a review of recent variants and applications." Neural Computing and Applications 30(2): 413-435.
- [4]. Feng, C.-W., et al. (2004). "Optimizing the schedule of dispatching RMC trucks through genetic algorithms." Automation in Construction 13(3): 327-340.
- [5]. Feng, C.-W. and H.-T. Wu (2006). "Integrating fmGA and CYCLONE to optimize the schedule of dispatching RMC trucks." Automation in Construction 15(2): 186-199.
- [6]. Jitkongchuen, D. (2015). A hybrid differential evolution with grey wolf optimizer for continuous global optimization. 2015 7th international conference on information technology and electrical engineering (ICITEE), IEEE.
- [7]. Kamboj, V. K. (2016). "A novel hybrid PSO-GWO approach for unit commitment problem." Neural Computing and Applications 27(6): 1643-1655.
- [8]. Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks, IEEE.
- [9]. Mirjalili, S. (2015). "The ant lion optimizer." Advances in engineering software 83: 80-98.
- [10]. Mirjalili, S. (2015). "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." Knowledge-based systems 89: 228-249.
- [11]. Mirjalili, S. (2016). "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems." Neural Computing and Applications 27(4): 1053-1073.
- [12]. Mirjalili, S. and A. Lewis (2016). "The whale optimization algorithm." Advances in engineering software 95: 51-67.
- [13]. Mirjalili, S., et al. (2014). "Grey wolf optimizer." Advances in engineering software 69: 46-61.
- [14]. Ntziachristos, L., et al. (2003). "Particle emissions characteristics of different on-road vehicles." SAE transactions: 1568-1578.

- [15]. Pan, J.-S., et al. (2017). A novel hybrid GWO-FPA algorithm for optimization applications. International conference on smart vehicular technology, transportation, communication and applications, Springer.
- [16]. Şenel, F. A., et al. (2019). "A novel hybrid PSO-GWO algorithm for optimization problems." Engineering with Computers 35(4): 1359-1373.
- [17]. Singh, N. and S. Singh (2017). "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance." Journal of Applied Mathematics 2017.
- [18]. Singh, N. and S. Singh (2017). "A novel hybrid GWO-SCA approach for optimization problems." Engineering Science and Technology, an International Journal 20(6): 1586-1601.
- [19]. Wolpert, D. H. and W. G. Macready (1997). "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1(1): 67-82.
- [20]. Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. International symposium on stochastic algorithms, Springer.
- [21]. Zhu, A., et al. (2015). "Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC." Journal of Systems Engineering and Electronics 26(2): 317-328.