

Áp dụng tối ưu sóng nước cho các bài toán tối ưu tổ hợp

Hoàng Bảo An - 22024545, Phan Tiến Đạt - 22024529

Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội

Ngày 12 tháng 8 năm 2025



- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết
- 7 Tài liệu tham khảo

Tổng quan về WWO

WWO là một kỹ thuật metaheuristic được phát triển dựa trên quan sát các sóng nước tự nhiên. Kỹ thuật này được liệt kê vào mục trí tuệ bầy đàn tự nhiên (Natural swarm-based) [1]

- Được giới thiệu lần đầu bởi Yu-Jun Zheng vào năm 2015 (Water wave optimization: A new nature-inspired metaheuristic, Computers & Operations Research - Elsevier) [2]. Mô phỏng hành vi lan truyền của sóng nước trong mô hình shallow water wave
- Ban đầu được áp dụng vào các bài toán tối ưu liên tục, nhưng sau đó được sự quan tâm và chú ý trong việc áp dụng với các bài toán tối ưu tổ hợp
- Năm 2019, Zheng và các cộng sự đã tổng hợp một số kỹ thuật và đánh giá WWO trong các bài toán tối ưu tổ hợp rời rạc [3]

WWO trong không gian liên tục

Mỗi lời giải x trong quần thể là một gợn sóng. Hàm mục tiêu đánh giá lời giải đó càng tốt thì bước sóng λ_x càng nhỏ (tương đương với việc năng lượng càng lớn, bước sóng càng nhỏ). Thuật toán bao gồm các toán tử cơ bản sau:

- Lan truyền
- Khúc xạ
- Vỡ sóng

Trong không gian liên tục, chiều thứ d của một lời giải \mathbf{x} sẽ được cập nhật như sau

$$\mathbf{x}'(d) = \mathbf{x}(d) + \lambda_{\mathbf{x}} \cdot \text{rand}(-1, 1) \cdot L(d) \quad (1)$$

Trong đó:

- rand là hàm ngẫu nhiên với phân bố đều
- $L(d)$ là khoảng tìm kiếm của chiều thứ d

Bước sóng của tất cả lời giải $\lambda_{\mathbf{x}}$ được khởi tạo bằng 0.5, sau đó sẽ được điều chỉnh dựa trên độ tốt của lời giải đó $f(\mathbf{x})$ qua các thể hệ.

$$\lambda_{\mathbf{x}} = \lambda_{\mathbf{x}} \cdot \alpha^{-(f(\mathbf{x}) - f_{\min} + \epsilon) / (f_{\max} - f_{\min} + \epsilon)} \quad (2)$$

Trong đó f_{\max} và f_{\min} là giá trị của lời giải tốt nhất và tệ nhất trong quần thể ở thế hệ hiện tại, α là hệ số suy giảm bước sóng (được đề xuất là 1.0026)[3].

Khúc xạ và vỡ sóng

- Khúc xạ: Nếu một gợn sóng (một lời giải) không được cải thiện qua một vài thế hệ thì sẽ bị loại bỏ, một lời giải mới - được tạo bằng cách lấy một điểm ngẫu nhiên nằm giữa lời giải cũ và lời giải tốt nhất \mathbf{x}^* - sẽ thay thế lời giải cũ.
- Vỡ sóng: Khi một lời giải tốt nhất \mathbf{x}^* được tìm thấy, nó sẽ được lưu lại, sau đó vỡ ra thành các lời giải con khác bằng cách di chuyển một khoảng nhỏ so với \mathbf{x}^* theo chiều ngẫu nhiên. Thực chất là ta sẽ áp dụng các kỹ thuật tìm kiếm cục bộ (local search) trong bước này.

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết
- 7 Tài liệu tham khảo

Các bước áp dụng vào trong bài toán tối ưu tổ hợp

Để áp dụng được WWO vào trong bài toán tối ưu tổ hợp, sẽ có 4 bước để rời rạc hóa các thuật toán, cụ thể như sau:

- Xác định biểu diễn của lời giải và cấu trúc lân cận của các lời giải để sử dụng local search
- Xác định toán tử lan truyền
- Xác định cách tính toán bước sóng sử dụng trong bước lan truyền
- Điều chỉnh lại thuật toán để thích nghi với các khía cạnh khác của thuật toán tối ưu rời rạc

Xác định toán tử lan truyền

Biến đổi \mathbf{x} với xác suất tỷ lệ thuận với bước sóng λ_x

Ví dụ với bài toán TSP: Gọi lời giải hiện tại là $\mathbf{x} = \{x_1, \dots, x_n\}$: Với mỗi i từ 1 đến n , với xác suất λ_x , đảo ngược một dãy con $\{x_i, \dots, x_{i+l}\}$ trong đó l là một số ngẫu nhiên nằm giữa $[1, n - i]$. Trong trường hợp này, λ_x phải là một số thực nằm giữa $[0, 1]$.

Biến đổi \mathbf{x} bằng cách thực hiện k bước tìm kiếm cục bộ

Với bài toán TSP, ta có thể thực hiện k lần hoán đổi hai thành phố ngẫu nhiên, với k là một số nguyên ngẫu nhiên trong khoảng $[1, \lambda_x]$. Trong trường hợp này, λ_x phải là một số nguyên nằm trong khoảng $[1, n]$.

Xác định cách tính toán bước sóng mới

Đặt $\lambda_{\mathbf{x}}$ tỉ lệ nghịch với độ tốt của lời giải $f(\mathbf{x})$:

Có hai dạng điển hình được sử dụng. Gọi \mathbf{P} là quần thể lời giải, λ_{max} và λ_{min} lần lượt là bước sóng cho phép lớn nhất và nhỏ nhất.

$$\lambda_{\mathbf{x}} = \lambda_{max} \frac{(\sum_{\mathbf{x}' \in \mathbf{P}} f(\mathbf{x}')) - f(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathbf{P}} f(\mathbf{x}')} \quad (3)$$

và

$$\lambda_{\mathbf{x}} = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \frac{f_{max} - f(\mathbf{x}) + \epsilon}{f_{max} - f_{min} + \epsilon} \quad (4)$$

Xác định cách tính toán bước sóng mới

Đặt $\lambda_{\mathbf{x}}$ tỉ lệ nghịch với $f(\mathbf{x})$ dựa trên mô hình hàm mũ.

$$\lambda_{\mathbf{x}} = \lambda_{min} \cdot b^{\alpha \cdot (f_{max} - f(\mathbf{x}) + \epsilon) / (f_{max} - f_{min} + \epsilon)} \quad (5)$$

trong đó b và α là hai tham số điều khiển. Để thuận tiện, α thường được đặt bằng 1, sau đó b được điều chỉnh theo $\lambda_{max}/\lambda_{min}$.

Cập nhật $\lambda_{\mathbf{x}}$ theo chiều nghịch đảo với sự thay đổi của $f(\mathbf{x})$

$$\lambda_{\mathbf{x}'} = \min\left(\lambda_{\mathbf{x}} + \alpha \frac{f(\mathbf{x}) - f(\mathbf{x}')}{f_{max}}, \lambda_{max}\right) \quad (6)$$

Ngoài các giá trị tốt nhất/tệ nhất của toàn quần thể trong một thế hệ, ta có thể theo dõi các giá trị tốt nhất/tệ nhất của một lời giải qua các thế hệ khác nhau.

Thích nghi với các khía cạnh khác của thuật toán tối ưu rời rạc

Toán tử vỡ sóng

Đặt n_b là một siêu tham số. Khi ta tìm được một lời giải tốt nhất, ta sẽ thực hiện local search để tạo ra n_b lời giải mới dựa trên lời giải cũ.

Ở bước vỡ sóng, tác giả đề xuất thực hiện các kỹ thuật local search sâu hơn. Khác với bước lan truyền, bước vỡ sóng sẽ tập trung vào khía cạnh khai thác (intensification/exploitation), còn bước lan truyền nên cân bằng cả hai yếu tố khai thác và khám phá (diversification/exploration).

Thích nghi với các khía cạnh khác của thuật toán tối ưu rời rạc

Cập nhật quần thể trong bước lan truyền

Khi cập nhật quần thể, một số kỹ thuật sau thường được áp dụng:

- Thay thế một lời giải \mathbf{x} bằng lời giải lan truyền \mathbf{x}' chỉ khi \mathbf{x}' tốt hơn.
- Thay thế một lời giải \mathbf{x} bằng lời giải lan truyền \mathbf{x}' chỉ khi \mathbf{x}' tốt hơn, hoặc $\exp((f(\mathbf{x}') - f(\mathbf{x}))/T)$ lớn hơn một ngưỡng nào đó. Ở đây, T là tham số nhiệt độ, giống trong thuật toán Simulated Annealing. Việc lai giữa WWO và SA được đề xuất trong [4]
- Thay thế một lời giải \mathbf{x} bằng lời giải lan truyền \mathbf{x}' nếu \mathbf{x}' tốt hơn hoặc \mathbf{x}' có thể cải thiện tính đa dạng của quần thể (khoảng cách giữa \mathbf{x}' và cá thể gần nhất của nó lớn hơn khoảng cách của \mathbf{x}).

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP**
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết
- 7 Tài liệu tham khảo

Áp dụng WWO vào bài toán SO-MKNAP

Phát biểu bài toán:

Cho một danh sách gồm n vật thể và m túi. Mỗi vật thể i ($1 \leq i \leq n$) có trọng lượng trong túi j ($1 \leq j \leq m$) là w_{ij} và có giá trị là v_i . Mỗi túi j có trọng lượng là c_j . Chọn ra một tập con các vật thể sao cho chúng vừa tất cả các túi và tổng giá trị là lớn nhất.

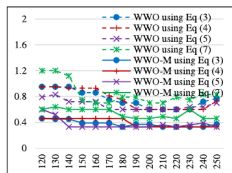
Ở bài toán này tác giả của bài báo đã sử dụng các phương pháp sau: Lời giải x sẽ là một vector gồm n giá trị $[0, 1]$. Toán tử lan truyền sẽ sử dụng phép lật bit. Phương trình 3, 4, 5 và 6 được sử dụng để cập nhật bước sóng qua từng thế hệ. Lời giải lan truyền nếu tốt hơn lời giải cũ thì lời giải cũ sẽ bị thay thế. Cuối cùng, số lượng lời giải trong quần thể sẽ bị giảm đi tuyến tính qua từng thế hệ, loại bỏ những lời giải tệ nhất ra khỏi quần thể. Ngoài ra, ở bước vỡ sóng, nhóm tác giả cũng đã thử nghiệm bằng cách linh động chọn ra một trong ba toán tử: loại bỏ lời giải tệ nhất thứ k ; thêm một vật phẩm có hệ số giá trị cao; và hoán đổi hai vật phẩm được chọn và chưa được chọn sao cho tổng giá trị là cao hơn.

Mã giả thuật toán Tối ưu sóng nước (WWO)

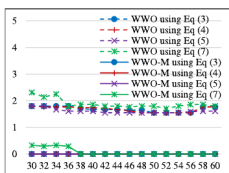
Algorithm 1 Tối ưu sóng nước (Water Wave Optimization)

```
1: Khởi tạo ngẫu nhiên một quần thể gồm  $NP$  lời giải cho bài toán;  
2: Đặt  $\mathbf{x}^*$  là lời giải tốt nhất trong quần thể;  
3: while điều kiện dừng chưa thỏa mãn do  
4:   Tính toán bước sóng  $\lambda$  cho mỗi lời giải;  
5:   for mỗi lời giải  $\mathbf{x}$  trong quần thể do  
6:      $k \leftarrow \text{rand}(1, \lambda_{\mathbf{x}})$ ;  
7:     Tạo lời giải mới  $\mathbf{x}'$  bằng cách đảo ngẫu nhiên  $k$  thành phần của  $\mathbf{x}$ ;  
8:     if  $f(\mathbf{x}') > f(\mathbf{x})$  then  
9:       Thay thế  $\mathbf{x}$  bằng  $\mathbf{x}'$  trong quần thể;  
10:      if  $f(\mathbf{x}) > f(\mathbf{x}^*)$  then  
11:        Thực hiện tìm kiếm cục bộ quanh  $\mathbf{x}$ ;  
12:        Cập nhật  $\mathbf{x}^*$  là lời giải tốt nhất giữa  $\mathbf{x}$  và các lân cận;  
13:      end if  
14:    end if  
15:  end for  
16:  Cập nhật kích thước quần thể;  
17: end while  
18: return  $\mathbf{x}^*$ 
```

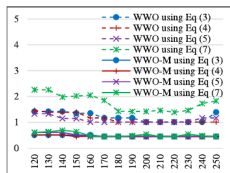
Kết quả thực nghiệm của tác giả



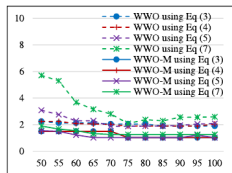
(d) instance 5 ($m = 5, n = 250$)



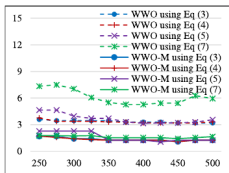
(e) instance 6 ($m = 30, n = 60$)



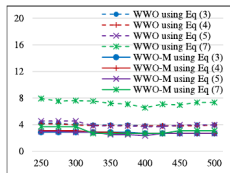
(f) instance 7 ($m = 10, n = 250$)



(g) instance 8 ($m = 30, n = 100$)



(h) instance 9 ($m = 10, n = 500$)



(i) instance 10 ($m = 30, n = 500$)

Hình: Sai lệch giữa kết quả tốt nhất theo từng thế hệ và optimal của instance (test case)[3]

Kết quả thực nghiệm của tác giả

Ins. ($n \times m$)		GA	PSO	FFO	WWO	ACO	HHS	BAAA	HGA	WWO-M
1 (2×28)	med	$\dagger 0.39$	0	0	0	0	0	0	0	0
	std	0.24	0	0	0	0	0	0	0	0
2 (2×105)	med	$\dagger 1.50$	0.36	0.32	0.32	$\dagger 0.25$	0	0	0	0
	std	0.39	0.25	0.30	0.44	0.22	0.1	0	0	0
3 (5×100)	med	$\dagger 0.85$	$\dagger 0.74$	0.37	0.55	$\dagger 0.02$	0	0	0	0
	std	0.53	0.17	0.26	0.33	0.14	0.12	0	0	0
4 (10×100)	med	$\dagger 1.93$	$\dagger 1.33$	1.27	1.31	$\dagger 1.18$	0.37	0.29	$\dagger 1.05$	0.39
	std	0.37	0.27	0.45	0.6	0.52	0.38	0.39	0.51	0.22
5 (5×250)	med	$\dagger 0.86$	$\dagger 0.95$	0.6	0.6	$\dagger 0.42$	-0.23	0.33	$\dagger 0.55$	0.33
	std	0.31	0.57	0.51	0.26	0.31	0.23	0.15	0.2	0.18
6 (30×60)	med	$\dagger 5.14$	-0.65	$\dagger 1.83$	1.55	0	0	0	0	0
	std	1.25	0.68	0.6	0.71	0.12	0	0	0.26	0
7 (10×250)	med	$\dagger 1.57$	$\dagger 1.89$	$\dagger 1.15$	1.01	$\dagger 1.08$	$\dagger 0.61$	0.48	$\dagger 1.14$	0.45
	std	0.82	0.7	0.72	0.51	0.36	0.33	0.28	0.35	0.25
8 (30×100)	med	$\dagger 3.61$	2.07	$\dagger 2.29$	1.9	$\dagger 1.88$	$\dagger 1.23$	1.03	$\dagger 1.67$	1.03
	std	2.21	1.57	1.43	0.99	1.02	0.35	0.24	0.4	0.21
9 (10×500)	med	$\dagger 3.73$	$\dagger 3.46$	3.31	3.2	$\dagger 3.09$	1.41	1.25	$\dagger 1.73$	1.07
	std	1.44	1.35	1.31	1.29	0.98	0.41	0.36	0.95	0.41
10 (30×500)	med	$\dagger 5.27$	$\dagger 4.66$	$\dagger 3.96$	3.83	$\dagger 3.71$	$\dagger 2.69$	2.43	$\dagger 2.8$	2.35
	std	1.98	1.03	0.89	0.7	0.69	0.6	0.46	0.68	0.48

Hình: So sánh giữa các thuật toán heuristic [3]

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack**
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết
- 7 Tài liệu tham khảo

Áp dụng WWO trong classic knapsack

Phát biểu bài toán:

Cho n đồ vật được đánh số từ 1 đến n , trong đó đồ vật thứ i có khối lượng w_i và giá trị v_i . Yêu cầu chọn một tập các đồ vật sao cho tổng khối lượng không vượt quá giới hạn W và tổng giá trị đạt lớn nhất có thể.

Nhược điểm của WWO gốc:

- **Propagation ngẫu nhiên, không định hướng:** dễ bỏ item giá trị cao hoặc thêm item chất lượng kém, gây nhiễu và chậm hội tụ.
- **Local search đơn giản:** thiếu các hoán đổi nhiều item \rightarrow dễ kẹt ở nghiệm cục bộ.
- **Xử lý nghiệm vi phạm kém:** dùng hàm phạt, nhiều bước lãng phí cho nghiệm infeasible.
- **Thiếu cơ chế đa dạng hoá:** không có chiến lược escape khi hội tụ sớm.
- **Không tận dụng đặc thù knapsack:** bỏ qua thông tin mật độ giá trị (density) để định hướng chọn/bỏ item.

Cải tiến 1: Propagation có định hướng theo điểm số

Lý do cần cải tiến:

- WWO gốc chỉ *lật ngẫu nhiên k bit* quanh nghiệm hiện tại \rightarrow dễ bỏ item tốt hoặc thêm item tệ \rightarrow lãng phí bước lan truyền.
- Knapsack: density là chỉ số mạnh để chọn/bỏ đúng.

Mô tả:

- Cải tiến: tính score cho từng item:
 - Item chưa chọn: score cao nếu density (v/w) lớn + thêm yếu tố giá trị.
 - Item đang chọn: score dựa trên nghịch đảo density (density thấp \rightarrow score cao).
- Chọn `max_candidates` item có score cao nhất bằng **heap** (tìm top-k hiệu quả).
- **Ngẫu nhiên hóa nhẹ** thứ tự các ứng viên để tránh cứng nhắc và duy trì đa dạng.
- Lật `num_changes` item từ danh sách ứng viên.
- Nếu nghiệm vi phạm ràng buộc \rightarrow gọi `repair_and_fill_solution`.

Khắc phục nhược điểm:

- Giảm nhiều ngẫu nhiên nhưng vẫn giữ một phần ngẫu nhiên để tránh kẹt local optima.
- Tăng xác suất cải thiện \rightarrow hội tụ nhanh, ổn định hơn.

Cải tiến 2: Local search giàu toán tử (1-1, 2-1, 1-2, 2-2)

Lý do cần cải tiến:

- LS gốc của WWO đơn giản, dễ kẹt ở local optima.
- Knapsack đôi khi cần bỏ nhiều item nhỏ để thêm item lớn (hoặc ngược lại).

Mô tả:

- Chọn **top-density unselected** (item chưa chọn tốt nhất) và **worst-density selected** (item đang chọn tệ nhất).
- Thử các hoán đổi:
 - 1-1: thay 1 xấu bằng 1 tốt.
 - 2-1: bỏ 2 xấu, thêm 1 tốt.
 - 1-2: bỏ 1 xấu, thêm 2 tốt.
 - 2-2: bỏ 2 xấu, thêm 2 tốt.

Khắc phục nhược điểm:

- Cho phép nhảy qua rào cản local optima.
- Khám phá không gian nghiệm rộng hơn, vẫn dựa trên density.

Cải tiến 3: Repair-then-fill khi quá tải

Lý do cần cải tiến:

- WWO gốc cho MKP dùng hàm phạt: nghiệm infeasible vẫn giữ \rightarrow lãng phí bước tìm kiếm.
- 1D knapsack: repair đơn giản và nhanh.

Mô tả:

- Nếu nghiệm vượt capacity:
 - Bỏ item tệ nhất (density thấp nhất) cho đến khi feasible.
 - Lắp đầy bằng item density cao nhất còn lại mà vẫn vừa capacity.

Khắc phục nhược điểm:

- Đảm bảo nghiệm feasible sớm.
- Fitness luôn phản ánh giá trị thật, không ảo do phạt.
- Giảm thời gian “chữa cháy” trong quá trình chạy.

Cải tiến 4: Đa dạng hoá khi trì trệ

Lý do cần cải tiến:

- WWO gốc dễ hội tụ sớm nếu quần thể bị hút vào 1 vùng nghiệm.
- Landscape của COP có nhiều “điểm chết”.

Mô tả:

- Dùng biến `stagnation_counter` đếm số vòng không cải thiện best.
- Nếu \geq ngưỡng: thay $\sim 20\%$ cá thể tệ nhất bằng nghiệm seed mới.

Khắc phục nhược điểm:

- Giữ đa dạng quần thể.
- Cung cấp hướng tìm mới khi bị kẹt.
- Kết hợp propagation định hướng để tìm nghiệm tốt hơn trong thời gian giới hạn.

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến**
- 6 Tổng kết
- 7 Tài liệu tham khảo

Giới thiệu Paper

Paper: *Energy management and power quality improvement of microgrid system through modified water wave optimization* (Energy Reports, 2023) [5]

Mục tiêu:

- Quản lý năng lượng (EMS) và cải thiện chất lượng điện năng (PQ) trong microgrid (MG) kết hợp: **PEM Fuel Cell, Pin Lithium-ion, Supercapacitor**.
- Tối ưu tham số bộ điều khiển PI để:
 - Nâng cao hiệu suất
 - Giảm tiêu thụ nhiên liệu
 - Tăng độ ổn định hệ thống

Phương pháp:

- Áp dụng **Water Wave Optimization (WWO)** để tự động tinh chỉnh tham số PI, đáp ứng tải nhanh, duy trì PQ ổn định
- Ưu điểm WWO: đơn giản, ít tham số, hiệu quả với quần thể nhỏ

Bài toán Tối ưu

Mục tiêu:

- Tối ưu **6 hệ số PI** cho 3 nguồn: Fuel Cell (FC), Battery (Bat), Supercapacitor (SC)
- Giảm **tổng tích phân bình phương sai số** điều khiển

Biến quyết định:

$$\mathbf{X} = [k_{p,FC}, k_{i,FC}, k_{p,Bat}, k_{i,Bat}, k_{p,SC}, k_{i,SC}]$$

Hàm mục tiêu:

$$\begin{aligned} J_{\text{total}} = & w_1 \int_0^T t \cdot |P_{\text{FC}}^{\text{ref}}(t) - P_{\text{FC}}(t)| dt + w_2 \int_0^T t \cdot |SOC_{\text{Bat}}^{\text{ref}}(t) - SOC_{\text{Bat}}(t)| dt \\ & + w_3 \int_0^T t \cdot |SOC_{\text{SC}}^{\text{ref}}(t) - SOC_{\text{SC}}(t)| dt \end{aligned} \quad (7)$$

Ràng buộc: giới hạn k_p, k_i ($0 \leq k_p \leq 10, 0 \leq k_i \leq 1$), SOC an toàn, công suất/duty cycle trong $[0, 1]$.

WWO trong bài báo

WWO gốc:

- Khởi tạo quần thể sóng P , bước sóng λ , chiều cao sóng h_{\max} , hệ số α , β , k_{\max} .
- Đánh giá hàm mục tiêu \rightarrow tìm nghiệm tốt nhất x^* .
- **Vòng lặp:**
 - *Propagation*: tạo nghiệm mới x' quanh x .
 - Nếu $f(x') > f(x)$: thay thế, có thể thực hiện *Breaking* nếu tốt hơn x^* .
 - Nếu không: giảm h , nếu $h = 0$ thì *Refraction* (dịch về gần x^*).
- Cập nhật λ và lặp lại tới khi dừng.

Nhược điểm:

- Mạnh khai thác cục bộ nhưng yếu khám phá toàn cục, dễ mất đa dạng và hội tụ sớm.
 - Quần thể cố định, khó cân bằng giữa exploration & exploitation.
- \Rightarrow MWWO thay đổi quy mô quần thể và nhịp bước sóng theo tiến trình tối ưu.

Cải tiến chính của MWWO

1) Hệ số “bước sóng” thích nghi (adaptive wavelength coefficient)

- **Vấn đề của WWO gốc:** quy tắc cập nhật bước sóng λ dựa vào hệ số α cố định. Khi α không đổi, mức “sải bước” ở pha dò tìm có thể không còn phù hợp theo thời gian, dễ mất đa dạng hoặc quá thô khi cần tinh chỉnh.
- **Cách MWWO sửa:** thay α cố định bằng α thích nghi theo thể hệ:

$$\alpha_i = \alpha_{\max} \left(\frac{i_{\text{MGN}} - i_{\text{CGN}} + 1}{i_{\text{MGN}}} \right)^{\gamma} \quad (8)$$

- i_{CGN} : chỉ số vòng lặp hiện tại (Current Generation Number).
- i_{MGN} : tổng số vòng lặp tối đa của thuật toán (Maximum Generation Number).
- γ : tham số điều chỉnh tốc độ giảm của α .
- Cơ chế này giúp dò tìm mạnh ở đầu (exploration) rồi thu hẹp sải bước để khai thác (exploitation) về cuối, giảm nguy cơ kẹt cục bộ trên các hàm đa cực trị.
- Trong thuật toán MWWO, việc cập nhật $\alpha \rightarrow \alpha_i$ được đặt thành bước riêng (Step 4) trước khi lan truyền (propagation).

2) Kích thước quần thể thích nghi (adaptive population size)

- **Vấn đề của WWO gốc:** giữ kích thước quần thể cố định qua mọi thế hệ \rightarrow kém linh hoạt, dễ kẹt cục bộ (premature convergence) khi phải cân bằng dò tìm rộng (exploration) và khai thác sâu (exploitation).
- **Cách MWWO sửa:** dùng một quy luật tuyến tính giảm dần quy mô quần thể theo số thế hệ:

$$n = n_{\max} - (n_{\max} - n_{\min}) \cdot \frac{i_{\text{CGN}}}{i_{\text{MGN}}} \quad (9)$$

- **Ý nghĩa:**
 - 1 Đầu quá trình \rightarrow quần thể lớn để quét rộng không gian nghiệm.
 - 2 Càng về sau \rightarrow quần thể nhỏ để tập trung tinh chỉnh cục bộ.
- **Lợi ích:** (i) cân bằng exploration/exploitation, giảm kẹt cục bộ; (ii) mỗi lần thu nhỏ quần thể thì giảm số lần đánh giá fitness, hạ chi phí tính toán mà vẫn cho nghiệm tốt hơn.

Kết quả so sánh PI – WWO – MWWO

Tiêu chí	PI	WWO	MWWO
Thời gian thực thi (s)	120	59	48
Tiêu thụ H ₂ (g/1000s)	50.50	41.39	40.81
Công suất tải FC (W)	1349	1580	2300
Hiệu suất tổng (%)	69.32	81.51	89.67
Stress FC	22.27	17.59	12.54
SoC pin (%)	70–52	70–55	70–59

Quan sát:

- MWWO vượt trội ở hầu hết tiêu chí: hiệu suất cao nhất, tiết kiệm nhiên liệu hơn, giảm stress thiết bị.
- Cải thiện rõ rệt chất lượng điện năng (PQ) và duy trì SoC pin ở mức an toàn hơn.

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết**
- 7 Tài liệu tham khảo

- **WWO** là thuật toán tối ưu metaheuristic lấy cảm hứng từ sóng nước, cấu trúc đơn giản nhưng khả năng thích nghi cao, áp dụng được cho cả bài toán liên tục và tổ hợp.
- Khi được **tùy biến theo đặc thù bài toán**, WWO có thể nâng cao hiệu suất tìm kiếm, tăng chất lượng nghiệm và tốc độ hội tụ.
- Các biến thể như **MWWO** cho thấy tiềm năng mở rộng, cân bằng tốt hơn giữa khai thác và khám phá, giảm nguy cơ hội tụ sớm.
- Thực nghiệm chứng minh WWO và các cải tiến mang lại **kết quả khả quan**, hứa hẹn ứng dụng hiệu quả cho nhiều bài toán tối ưu phức tạp trong tương lai.

Mục lục

- 1 Tổng quan
- 2 Các kỹ thuật áp dụng cho tối ưu tổ hợp
- 3 Áp dụng vào SO-MKP
- 4 Áp dụng trong classic knapsack
- 5 Bài báo liên quan & cải tiến
- 6 Tổng kết
- 7 Tài liệu tham khảo**

- [1] Kashif Hussain, Mohd Najib Mohd Salleh, Shi Cheng, and Yuhui Shi.
Metaheuristic research: a comprehensive survey.
Artificial intelligence review, 52(4):2191–2233, 2019.
- [2] Yu-Jun Zheng.
Water wave optimization: a new nature-inspired metaheuristic.
Computers & Operations Research, 55:1–11, 2015.
- [3] Yu-Jun Zheng, Xue-Qin Lu, Yi-Chen Du, Yu Xue, and Wei-Guo Sheng.
Water wave optimization for combinatorial optimization: Design strategies and applications.
Applied Soft Computing, 83:105611, 2019.
- [4] WL Wang, C Chen, Li Li, and WK Li.
Adaptive water wave optimization algorithm based on simulated annealing.
Comput. Sci., 44(10):216–227, 2017.

- [5] Subhashree Choudhury, George Tom Varghese, Satyajit Mohanty, Venkata Ratnam Kolluru, Mohit Bajaj, Vojtech Blazek, Lukas Prokop, and Stanislav Misak. Energy management and power quality improvement of microgrid system through modified water wave optimization. *Energy Reports*, 9:6020–6041, 2023.