Tạp chí Khoa học Đại học Cần Thơ
Số chuyên đề: Khoa học tự nhiên
ISSN 1859-2333 | e-ISSN 2815-5599

DOI:10.22144/ctujos.2024.321

ỨNG DỤNG CÁC THUẬT TOÁN NATURE-INSPIRED VÀO BÀI TOÁN *P*-MEDIAN TRÊN MẶT PHẲNG

Nguyễn Ngọc Đăng Duy[1*] và Nguyễn Hà Công Lý[2]
[1]*Khoa Khoa học Tự nhiên, Trường Đại học Cần Thơ*
[2]*Bộ môn Toán, Trường Đại học FPT Cần Thơ*
**Tác giả liên hệ (Corresponding author): duy3300@gmail.com*

Thông tin chung (Article Information)

Nhận bài (Received): 17/04/2024
Sửa bài (Revised): 06/06/2024
Duyệt đăng (Accepted): 17/07/2024

Title: *Application of nature-inspired algorithms to the p-median problem on the plane*

Author(s): *Nguyen Ngoc Dang Duy[1*] and Nguyen Ha Cong Ly[2]*

Affiliation(s): *[1]Can Tho University, [2]FPT University - Cantho Campus*

TÓM TẮT

Trong bài báo này, bài toán p-median trên mặt phẳng được đề cập với chuẩn Euclide. Bên cạnh đó, một số thuật giải heuristic thường dùng như thuật toán Tối ưu bầy đàn (Particle Swarm Optimization-PSO), thuật toán Bầy sói xám (Grey Wolf Optimizer-GWO), thuật toán Tối ưu đàn dơi (Bat Algorithm-BA) và thuật toán Tối ưu bầy mèo (Cat Swarm Optimization-CSO) được sử dụng để tìm ra nghiệm gần đúng cho bài toán p-median trên mặt phẳng.

Từ khoá: *Bài toán Fermat-Weber, bài toán p-median, thuật giải heuristic, thuật toán nature-inspired*

ABSTRACT

In this article, the p-median problem on a plane is discussed with the Euclidean norm. Additionally, several commonly used heuristic algorithms such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Bat Algorithm (BA), and Cat Swarm Optimization (CSO) are employed to find approximate solutions for the p-median problem on the plane.

Keywords: *Fermat-Weber problem, p-median problem, heuristic algorithm, nature-inspired algorithm*

1. PHẦN MỞ ĐẦU

Bài toán vị trí từ lâu đã trở thành một bài toán đóng vai trò quan trọng trong đa dạng các lĩnh vực, đặc biệt là trong lĩnh vực tối ưu tổ hợp và ứng dụng vào các mô hình kinh tế. Chính vì lí do đó mà mô hình bài toán vị trí được quan tâm nghiên cứu bởi các nhà toán học trên khắp thế giới.

Một số nghiên cứu gần đây đối với bài toán *p*-median có thể kể đến như Mauricio and Renato (2004) đã đề xuất một thuật toán heuristic để tìm các giải pháp gần tối ưu cho bài toán *p*-median. Chang et al. (2016) đã nghiên cứu bài toán liên thông *p*-median trên đồ thị khối và giải bài toán này bằng một thuật toán thời gian tuyến tính. Duy và Hiếu (2020) đã nghiên cứu tìm lời giải cho bài toán liên thông *p*-median trên đồ thị đầy đủ và đồ thị lưỡng phân đầy đủ. Đồng thời, các tác giả đề xuất thuật toán thời gian tuyến tính cho bài toán này. Kien et al. (2021) đề xuất một thuật toán thời gian $O(n\log n)$ cho bài toán liên thông *p*-median trên đồ thị đa lớp đầy đủ.

Bài báo này tìm nghiệm gần đúng cho bài toán tối ưu *p*-median trên mặt phẳng bằng các thuật toán heuristic thường gặp như thuật toán Tối ưu bầy đàn (PSO), thuật toán Bầy sói xám (GWO), thuật toán Tối ưu đàn dơi (BA) và thuật toán Tối ưu bầy mèo (CSO).

2. BÀI TOÁN *P*-MEDIAN TRÊN MẶT PHẲNG

2.1. Bài toán Fermat-Weber

Được biết đến lần đầu tiên vào khoảng thế kỷ 17, bài toán lúc đó được phát biểu bởi Pierre de Fermat. Mục tiêu của bài toán là tìm một điểm mới trên mặt phẳng sao cho tổng khoảng cách từ điểm đó đến ba đỉnh của tam giác cho trước là nhỏ nhất. Bài toán này sau đó đã được giải bởi Evangelista Torricelli bằng hình học và nghiệm của bài toán này còn được gọi là điểm Torricelli trong tam giác. Về sau, nhà kinh tế học người Đức, Alfred Weber đã tổng quát bài toán lên thành trường hợp nhiều điểm cho trước trên mặt phẳng và mỗi điểm được gắn với một trọng số nhằm mục đích mang lại hiệu quả kinh tế. Kể từ đó, bài toán được mang tên bài toán Fermat-Weber trên mặt phẳng, có thể phát biểu lại như sau: Cho trước một tập hợp chứa hữu hạn một số điểm $P_i(i=1,2,\ldots,m)$ trong không gian Euclide hữu hạn chiều $\mathbb{R}^n$. Mỗi đỉnh được gắn với một trọng số tương ứng $w_i(i=1,2,\ldots,m)$. Tìm điểm X để hàm mục tiêu sau đây đạt giá trị nhỏ nhất

$$f(X)=\sum_{i=1}^{m} w_i \| X-P_i \|.$$

Bài báo này sử dụng $\| X-Y \|=\sqrt{(x_1-y_1)^2+(x_1-y_2)^2}$ với $X=(x_1,x_1), Y=(y_1,y_2)\in\mathbf{R}^2$.

2.2. Bài toán *p*-median và một số tính chất

Trong mặt phẳng, cho trước tập hợp gồm n điểm phân biệt $\mathcal{P}=\{P_1,P_2,\ldots,P_n\}$ với tập trọng số tương ứng $\mathcal{W}=\{w_1,w_2,\ldots,w_n\}$. Bài toán p-median trên mặt phẳng là đi tìm một tập hợp $\mathcal{X}=\left\{X_1,X_2,\ldots,X_p\right\}$ gồm p điểm phân biệt sao cho hàm mục tiêu

$$f(\mathcal{X})=\sum_{i=1}^{n} w_i \min\left\{d(X_j,P_i)\mid j=1,\ldots,p\right\}$$

đạt giá trị nhỏ nhất. Trong đó $d(X,Y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$ với $X,Y\in\mathbf{R}^2$.

Đầu tiên, ta xét bài toán *p*-median với $p=1$. Khi đó, mục tiêu của bài toán là đi tìm một điểm X sao cho hàm mục tiêu $f(X)=\sum_{i=1}^{n} w_i d(X,P_i)$ đạt giá trị nhỏ nhất. Dễ thấy đây chính là bài toán Fermat-Weber tổng quát. Năm 1937, nhà toán học người Hungary, Endre Weiszfeld đã đề xuất một thuật toán giải gần đúng nghiệm của bài toán Fermat-Weber tổng quát, Beck and Sabach (2015) trình bày thuật toán nêu trên và một số kết quả mới có liên quan. Bây giờ với $p=2$, bài toán *p*-median là đi tìm một tập hợp gồm 2 điểm $\mathcal{X}=\left\{X_1,X_2\right\}$ sao cho hàm mục tiêu $f(\mathcal{X})=\sum_{i=1}^{n} w_i \min\left\{d(X_j,P_i)\mid j=1,2\right\}$ đạt giá trị nhỏ nhất. Để ý rằng, với tập hợp $\mathcal{X}^*=\left\{X_1,X_2\right\}$ là tập hợp chứa điểm tối ưu 2-median, ta luôn có thể chia tập hợp các điểm có sẵn thành

$$\mathcal{P}^{\leqslant}=\left\{P_i\in\mathcal{P}: w_i d(X_1,P_i)\leqslant w_i d(X_2,P_i)\right\}$$

$$\mathcal{P}^{>}=\left\{P_i\in\mathcal{P}: w_i d(X_1,P_i)> w_i d(X_2,P_i)\right\}.$$

Trong đó $\mathcal{P}^{\leqslant}$ là tập hợp chứa các đỉnh P_i sao cho khoảng cách có trọng số từ đỉnh này đến X_1 không vượt quá X_2 và $\mathcal{P}^{>}$ là tập hợp chứa các đỉnh P_i sao cho khoảng cách có trọng số từ đỉnh này đến X_1 lớn hơn X_2.

Từ đây, ta có được Bổ đề 2.2.1.

Bổ đề 2.2.1 Nếu $\mathcal{X}^*=\left\{X_1,X_2\right\}$ là tập hợp chứa điểm tối ưu 2-median thì X_1 là 1-median của tập hợp $\mathcal{P}^{\leqslant}$ và X_2 là 1-median của $\mathcal{P}^{>}$.

Chứng minh

Giả sử $\mathcal{X}^*=\left\{X_1,X_2\right\}$ là 2-median, khi đó ta xét các trường hợp sau:

1) X_1 không là 1-median của $\mathcal{P}^{\leqslant}$, X_2 không là 1-median của $\mathcal{P}^{>}$.

Xét $X'=\left\{X_{1'},X_{2'}\right\}$ với $X_{1'}$ là 1-median của $\mathcal{P}^{\leqslant}$ và $X_{2'}$ là 1-median của $\mathcal{P}^{>}$, khi đó ta có

$$\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P) > \sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)$$

$$\sum_{P\in\mathcal{P}^{>}} w_P d(X_2,P) > \sum_{P\in\mathcal{P}^{>}} w_P d(X_2',P).$$

Do đó

$$\begin{aligned}
&f(\mathcal{X}')-f(\mathcal{X}^*)\\
&=\sum_{i=1}^{n} w_i \min\{d(X_{1'},P_i),d(X_{2'},P_i)\}\\
&\quad-\sum_{i=1}^{n} w_i \min\{d(X_1,P_i),d(X_2,P_i)\}\\
&=\sum_{P\in\mathcal{P}^{\leqslant}} w_P \min\{d(X_{1'},P),d(X_{2'},P)\}\\
&\quad+\sum_{P\in\mathcal{P}^{>}} w_P \min\{d(X_{1'},P),d(X_{2'},P)\}\\
&\quad-\sum_{i=1}^{n} w_i \min\{d(X_1,P_i),d(X_2,P_i)\}\\
&\leqslant\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)+\sum_{P\in\mathcal{P}^{>}} w_P d(X_2',P)\\
&\quad-\left(\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P)+\sum_{P\in\mathcal{P}^{>}} w_P d(X_2,P)\right)\\
&=\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)-\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P)\\
&\quad+\sum_{P\in\mathcal{P}^{>}} w_P d(X_2',P)-\sum_{P\in\mathcal{P}^{>}} w_P d(X_2,P)<0.
\end{aligned}$$

Nên từ đây ta có $f(\mathcal{X}')<f(\mathcal{X}^*)$ (vô lí).

2) Nếu X_1 không là 1-median của $\mathcal{P}^{\leqslant}$ và X_2 là 1-median của $\mathcal{P}^{>}$.

Xét $X'=\{X_{1'},X_{2'}\}$ với $X_{1'}$ là 1-median của $\mathcal{P}^{\leqslant}$ và $X_{2'}$ là 1-median của $\mathcal{P}^{>}$, khi đó ta có

$$\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P) > \sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)$$

$$\sum_{P\in\mathcal{P}^{>}} w_P d(X_2,P) = \sum_{P\in\mathcal{P}^{>}} w_P d(X_2',P).$$

Do đó

$$\begin{aligned}
&f(\mathcal{X}')-f(\mathcal{X}^*)\\
&=\sum_{i=1}^{n} w_i \min\{d(X_{1'},P_i),d(X_{2'},P_i)\}\\
&\quad-\sum_{i=1}^{n} w_i \min\{d(X_1,P_i),d(X_2,P_i)\}\\
&=\sum_{P\in\mathcal{P}^{\leqslant}} w_P \min\{d(X_{1'},P),d(X_{2'},P)\}\\
&\quad+\sum_{P\in\mathcal{P}^{>}} w_P \min\{d(X_{1'},P),d(X_{2'},P)\}\\
&\quad-\sum_{i=1}^{n} w_i \min\{d(X_1,P_i),d(X_2,P_i)\}\\
&\leqslant\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)+\sum_{P\in\mathcal{P}^{>}} w_P d(X_2',P)\\
&\quad-\left(\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P)+\sum_{P\in\mathcal{P}^{>}} w_P d(X_2,P)\right)\\
&=\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1',P)-\sum_{P\in\mathcal{P}^{\leqslant}} w_P d(X_1,P)<0.
\end{aligned}$$

nên từ đây ta có $f(\mathcal{X}')<f(\mathcal{X}^*)$ (vô lí).

3) Nếu X_1 là 1-median của $\mathcal{P}^{\leqslant}$ và X_2 không là 1-median của $\mathcal{P}^{>}$.

Chứng minh tương tự như 2), ta cũng có điều vô lí.

Như vậy bổ đề được chứng minh.

3. MỘT SỐ THUẬT TOÁN NATURE-INSPIRED THƯỜNG GẶP

3.1. Thuật toán Tối ưu bầy đàn

Kennedy và Eberhart (1995) công bố một thuật toán tối ưu mang tên Tối ưu bầy đàn (Particle Swarm Optimization - PSO). Thuật toán lấy cảm hứng từ cách thức hoạt động trong quá trình tìm kiếm thức ăn của các quần thể động vật trong tự nhiên, đặc biệt là sự tương tác giữa các cá thể trong cùng một bầy đàn. Quá trình diễn ra thuật toán được mô tả như trong Thuật toán 1.

Thuật toán 1: Tối ưu bầy đàn (PSO)

Input: Bài toán tối ưu hàm mục tiêu $f(\cdot)$.

Bước 1: Khởi tạo

Các hằng số $c_1,c_2,k_{\max}$, r_1,r_2.

Ngẫu nhiên n vị trí và vận tốc của các cá thể trong bầy đàn.

Chọn ngẫu nhiên các hằng số quán tính w trên $(0;1]$.

Đặt $k=1$.

Bước 2: Vòng lặp tối ưu

Cập nhật các giá trị $P_{i,best}^{k}(i=1,\ldots,n)$.

While $k \leqslant k_{\max}$

Cập nhật vận tốc của các cá thể.

Cập nhật vị trí mới của các cá thể.

Cập nhật $P_{i,best}^{k}$ đối với mỗi cá thể

$$P_{i,best}^{k}=\begin{cases} P_{i,best}^{k-1},\ f(x_i^k)>f(P_{i,best}^{k-1}). \\ x_i^k\ \ ,f(x_i^{k-1})\leqslant f(P_{i,best}^{k-1}). \end{cases}$$

Cập nhật vị trí tối ưu $G_{best}^{k}=\arg\min\{f(P_{i,best}^{k})\}$

$$k:=k+1$$

Endwhile

Output: Vị trí tối ưu G_g^k và giá trị hàm tối ưu $f(G_g^k)$.

Các tham số có thể điều chỉnh trong thuật toán bao gồm:

1) Số lượng cá thể trong bầy đàn.

2) c_1 :Ảnh hưởng của vị trí tốt nhất mà mỗi cá thể tìm được.

3) c_2 : Ảnh hưởng của vị trí tốt nhất trên cả bầy đàn.

4) k_{max} : Số vòng lặp tối đa để chạy thuật toán.

5) w: Hệ số quán tính của mỗi cá thể.

3.2. Thuật toán Bầy sói xám

Thuật toán Bầy sói xám (Grey Wolf Optimizer - GWO) được nghiên cứu và phát triển bởi Mirjalili et al. (2014). Thuật toán lấy ý tưởng từ việc hợp tác trong quá trình săn tìm mồi của bầy sói trong tự nhiên. Quá trình diễn ra thuật toán được mô tả trong Thuật toán 2.

Thuật toán 2: Bầy sói xám (GWO)

Input: Bài toán tối ưu hàm mục tiêu $f(\cdot)$.

Khởi tạo bầy sói $X_i(i=1,\ldots,n)$

Khởi tạo các hằng số a, A, C

Số vòng lặp tối đa $t_{\max}$

Tính toán giá trị hàm mục tiêu đối với mỗi cá thể trong bầy sói

X_α là vị trí của sói có giá trị hàm tốt nhất

X_β là vị trí của sói có giá trị hàm tốt thứ hai

X_δ là vị trí của sói có giá trị hàm tốt thứ ba

While $t<t_{\max}$

Cập nhật vị trí của các cá thể sói trong bầy đàn theo công thức (1)

Cập nhật các hằng số a, A, C

Tính toán giá trị hàm mục tiêu đối với mỗi cá thể trong bầy sói

Cập nhật giá trị $X_\alpha, X_\beta, X_\delta$.

Endwhile

Output: Vị trí tối ưu X_α và giá trị hàm tối ưu $f(X_\alpha)$.

$$D_\alpha=\left|C_1\overline{X_\alpha}-\overline{X}\right|;\ D_\beta=\left|C_2\overline{X_\beta}-\overline{X}\right|;$$
$$D_\delta=\left|C_3\overline{X_\delta}-\overline{X}\right|$$

$$\overrightarrow{X_1}=\overrightarrow{X_\alpha}-D_\alpha\overrightarrow{A_1};\ \overrightarrow{X_2}=\overrightarrow{X_\beta}-D_\beta\overrightarrow{A_2};$$
$$\overrightarrow{X_3}=\overrightarrow{X_\delta}-D_\delta\overrightarrow{A_3}$$

$$\vec{X}(t+1)=\frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \qquad (1)$$

Các tham số có thể điều chỉnh của thuật toán này bao gồm:

1) Số lượng sói trong bầy.

2) $t_{\max}$: Số lần lặp tối đa mà thuật toán sẽ thực hiện tìm kiếm.

3) a: Trong thuật toán, ta có thiết lập $\vec{A} = 2\vec{a}r_1 - \vec{a}$, trong đó các thành phần của $\vec{a}$ giảm tuyến tính từ 2 về 0 và r_1 là đại lượng ngẫu nhiên nhận giá trị trên đoạn $[0,1]$.

4) C: Trong thuật toán, ta có thiết lập $C = 2r_2$ với r_2 là đại lượng ngẫu nhiên nhận giá trị trên đoạn $[0,1]$.

3.3. Thuật toán Tối ưu đàn dơi

Yang (2010) đã giới thiệu một thuật toán heuristic dựa trên hoạt động của đàn dơi (Bat Algorithm – BA), lấy ý tưởng từ sự di chuyển và truyền tín hiệu thông qua sóng âm giữa các cá thể trong một bầy. Thuật toán được mô tả chi tiết như trong Thuật toán 3.

Thuật toán 3: Tối ưu đàn dơi (BA)

Input: Bài toán tối ưu hàm mục tiêu $f(\cdot)$.

Khởi tạo quần thể dơi $x_i (i = 1,\ldots,n)$ và vận tốc v_i.

Khởi tạo tần số âm f_i với mỗi x_i.

Khởi tạo tần số xung r_i và độ vang A_i.

Số vòng lặp tối đa $t_{\max}$

While $t < t_{\max}$

Khởi tạo ngẫu nhiên các vị trí mới x_i bằng cách điều chỉnh tần số và cập nhật vận tốc, vị trí của các cá thể trong bầy dơi (công thức (2) đến (4))

If $rand > r_i$

Chọn một giải pháp tốt nhất từ các giải pháp của các cá thể trong bầy đạt được

Tạo ra một giải pháp cục bộ xung quanh giải pháp tốt nhất được chọn

Endif

Tạo ra một giải pháp mới bằng cách bay ngẫu nhiên

If $rand < A_i$ **and** $f(x_i) < f(x^*)$

Chấp nhận các giải pháp mới

Tăng r_i và giảm A_i

Endif

Xếp hạng các con dơi và tìm ra giải pháp tốt nhất hiện tại x^*

Endwhile

Output: Giải pháp tối ưu

$$f_i = f_{min} + \left(f_{max} - f_{min}\right)\beta \quad \text{với } \beta \in [0,1] \tag{2}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x^*\right) f_i \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

Một số tham số có thể điều chỉnh trong thuật toán Tối ưu bầy dơi bao gồm:

1) Số lượng dơi trong bầy.

2) A_i (Loudness): Độ vang, thể hiện thông qua giá trị âm thanh của dơi. Trong thuật toán, thường được thiết lập $A_i^{t+1} = \alpha A_i^t$ với t là số vòng lặp hiện tại (thường chọn $\alpha = 0,9$).

3) r_i (Rate of pulse emission): Tần suất mà dơi thay đổi vận tốc của mình, thường được thiết lập $r_i^{t+1} = r_i^0\left(1 - e^{-\gamma t}\right)$ với t là số vòng lặp hiện tại (thường chọn $\gamma = 0,9$).

4) Số lần lặp tối đa : Số lần lặp tối đa mà thuật toán sẽ thực hiện tìm kiếm.

3.4. Thuật toán Tối ưu bầy mèo

Một nhóm các nhà nghiên cứu gồm Chu et al. (2006) đã đề xuất một thuật toán heuristic dựa trên hành vi của các cá thể mèo trong một bầy mèo (Cat Swarm Optimization - CSO). Các cá thể mèo dành phần lớn thời gian nghỉ ngơi mỗi ngày, tuy nhiên tính cảnh giác của chúng rất nhạy bén. Chúng luôn cảnh giác với mọi tiếng động xung quanh bản thân kể cả khi đang trong trạng thái nghỉ ngơi. Mặc dù trông có vẻ lười biếng, nhưng các mèo rất thông minh và hành động rất cẩn thận. Thuật toán Tối ưu bầy mèo được xây dựng dựa trên hai trạng thái chính của một con mèo, bao gồm chế độ Tìm kiếm (Seeking mode) và chế độ Theo dõi (Tracing mode).

Để xây dựng hai chế độ này, đầu tiên chúng ta định nghĩa các tham số có thể điều chỉnh để tối ưu hiệu suất của thuật toán:

1) SMP (seeking memory pool): SMP được sử dụng để xác định kích thước của bộ nhớ tìm kiếm

cho mỗi con mèo, đại diện cho các điểm mà con mèo đang tìm kiếm. Con mèo sẽ chọn một điểm từ bộ nhớ theo các quy tắc được mô tả sau.

2) SRD (seeking range of the selected dimension): SRD xác định tỷ lệ biến đổi cho các chiều được chọn. Trong chế độ tìm kiếm, nếu một chiều được chọn để biến đổi, sự khác biệt giữa giá trị mới và giá trị cũ sẽ không vượt quá khoảng, được định nghĩa bởi SRD.

3) CDC (counts of dimension to change): CDC tiết lộ bao nhiêu chiều sẽ thay đổi. Những yếu tố này đều đóng vai trò quan trọng trong chế độ tìm kiếm.

4) MR (mixture ratio): Tỷ lệ mèo được chọn để tham gia chế độ theo dõi trong mỗi vòng lặp.

5) SPC (self-position considering): SPC là một biến Boolean, quyết định liệu điểm mà con mèo đang đứng có phải là một trong những vị trí khả thi để di chuyển đến không. Bất kể giá trị của SPC là đúng hay sai; giá trị của SMP sẽ không bị ảnh hưởng.

6) Số vòng lặp tối đa.

7) Số lượng mèo trong bầy.

Chế độ Tìm kiếm được thiết lập như sau:

Bước 1: Tạo j bản sao của vị trí hiện tại của mèo thứ k (cat_k), trong đó $j = \text{SMP}$. Nếu giá trị của SPC là đúng (True), ta đặt $j = \text{SMP} - 1$, sau đó giữ nguyên vị trí hiện tại như một trong những giá trị khả thi.

Bước 2: Đối với mỗi bản sao, dựa trên giá trị CDC, ngẫu nhiên cộng hoặc trừ SRD phần trăm của các giá trị tối ưu hiện tại và thay thế cho các giá trị cũ.

Bước 3: Tính toán giá trị thích nghi (FS) của tất cả các điểm khả thi.

Bước 4: Nếu tất cả FS không bằng nhau, tính toán xác suất lựa chọn của mỗi điểm khả thi theo công thức

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}.$$

Nếu ngược lại, ta đặt tất cả xác suất lựa chọn của mỗi điểm khả thi là 1.

Bước 5: Ngẫu nhiên chọn một điểm để di chuyển từ các điểm khả thi và thay thế vị trí của mèo thứ *k*.

Nếu mục tiêu của bài toán tối ưu là nghiệm tối ưu nhỏ nhất thì ta chọn $FS_b = FS_{max}$, ngược lại thì ta chọn $FS_b = FS_{min}$.

Bên cạnh đó, chế độ Theo dõi được thiết lập như sau:

Bước 1: Cập nhật vận tốc $v_{k,d}$ với mỗi chiều không gian dựa trên công thức

$$v_{k,d} = v_{k,d} + r_1 c_1 \left(x_{best,d} - x_{k,d} \right)$$

với $d = 1, \ldots, M$ (M là số chiều không gian đối với bài toán đang xét), c_1 là hằng số và r_1 là hằng số nhận giá trị ngẫu nhiên trên $[0,1]$.

Bước 2: Kiểm tra xem tốc độ có nằm trong khoảng tốc độ tối đa không. Trong trường hợp tốc độ mới vượt quá giới hạn, đặt nó bằng giới hạn.

Bước 3: Cập nhật vị trí của mèo thứ k theo công thức

$$x_{k,d} = x_{k,d} + v_{k,d}$$

Dựa trên hai chế độ này, quá trình diễn ra thuật toán Tối ưu bầy mèo được mô tả như trong Thuật toán 4.

Thuật toán 4: Tối ưu bầy mèo (CSO)

Input: Bài toán tối ưu hàm mục tiêu $f(\cdot)$.

Bước 1: Tạo ngẫu nhiên N con mèo.

Bước 2: Đặt ngẫu nhiên các con mèo vào không gian tìm kiếm có M chiều và ngẫu nhiên chọn các giá trị nằm trong khoảng tốc độ tối đa để đặt vào tốc độ của mỗi con mèo. Sau đó, ngẫu nhiên chọn một số con mèo và đặt chúng vào chế độ theo dõi theo tham số MR, còn lại đặt vào chế độ tìm kiếm.

Bước 3: Đánh giá giá trị thích nghi của mỗi con mèo bằng cách tính giá trị hàm mục tiêu ứng với vị trí của mỗi mèo và giữ lại con mèo tốt nhất trong bộ nhớ. Chú ý rằng chỉ cần ghi nhớ vị trí của con mèo tốt nhất (x_{best}) vì nó đại diện cho giải pháp tốt nhất cho đến hiện tại.

Bước 4: Di chuyển các con mèo theo các trạng thái của chúng. Nếu con mèo thứ k ở chế độ tìm kiếm, áp dụng con mèo vào quá trình tìm kiếm, nếu ngược lại thì áp dụng nó vào quá trình theo dõi. Các bước quá trình được trình bày ở trên.

Bước 5: Chọn lại một số con mèo và đặt chúng vào chế độ theo dõi theo tham số MR, sau đó đặt các con mèo còn lại vào chế độ tìm kiếm.

Bước 6: Kiểm tra điều kiện dừng thuật toán, nếu thoả mãn, kết thúc chương trình, ngược lại lặp lại từ **Bước 3** đến **Bước 5**.

Output: Vị trí tối ưu x_{best} và giá trị hàm tối ưu $f(x_{best})$.

4. TÍNH TOÁN TRÊN BỘ DỮ LIỆU ĐƯỢC TẠO NGẪU NHIÊN

Trong bài báo này, thuật toán Tối ưu bầy đàn (PSO), thuật toán Bầy sói xám (GWO), thuật toán Tối ưu đàn dơi (BA), thuật toán Tối ưu hóa bầy mèo (CSO) được sử dụng để giải gần đúng bài toán p-median trên mặt phẳng. Đối với các thuật toán này, hai điều kiện dừng được thiết lập để so sánh trực quan quá trình xử lí của thuật toán. Điều kiện dừng thứ nhất sử dụng sai số tuyệt đối của giá trị hàm mục tiêu tối ưu sau mỗi vòng lặp. Cụ thể

$$\text{relative change} = |\text{new solution - previous solution}|$$

trong đó new solution là giá trị hàm tối ưu ở vòng lặp hiện tại và previous solution là giá trị hàm tối ưu ở vòng lặp trước đó. Thuật toán dừng khi $\text{realtive change} < 0,001$. Điều kiện dừng thứ hai là số lượng vòng lặp đạt tới giới hạn cho trước là 200 vòng lặp.

Để so sánh khả năng xử lí của các thuật toán, bộ dữ liệu về tọa độ điểm có sẵn trên mặt phẳng được tạo ngẫu nhiên $P_i(x_i, y_i) \sim U(0,200) \times U(0,200)$, với $i = 1,\ldots,n$ và $U(a,b)$ là hàm ngẫu nhiên phân phối đều trên khoảng (a,b). Đối với trọng số đỉnh, khởi tạo ngẫu nhiên $w_i \sim U\left(0, \frac{n}{8}\right)$ được sử dụng với n là số lượng điểm cơ sở.

Bằng khởi tạo này, chúng tôi có ngẫu nhiên 5 bộ điểm cơ sở với số lượng điểm 50, 100, 200, 500, 1000 và ứng với mỗi bộ điểm, chúng tôi sử dụng cùng một bộ dữ liệu và chạy cả 4 thuật toán để có kết quả cụ thể như trong Bảng 1 đến Bảng 4.

Bảng 1. Kết quả chạy thuật toán PSO

Số lượng điểm cơ sở	p	Điều kiện dừng 200 vòng lặp		Điều kiện dừng $\text{realtive change} < 0,001$	
		Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình	Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình
50	10	0,617	2127,819	0,012	3022,702
	20	1,169	1435,015	0,022	2510,332
	30	1,738	1008,660	0,053	1804,781
	40	2,207	855,585	0,053	1732,768
100	20	2,249	7080,520	0,083	9934,153
	40	4,323	4417,565	0,044	7188,147
	60	6,460	3544,859	0,138	5955,358
	80	8,655	2935,293	0,129	4943,300
200	40	12,816	20598,833	0,378	28624,255
	80	24,200	14013,577	0,761	20462,685
	120	32,248	12153,630	0,918	16966,448
	160	34,044	10006,627	0,750	14357,155
500	100	71,443	91347,246	0,715	121062,471
	200	105,465	66575,881	2,640	86112,031
	300	156,211	54149,908	5,216	68898,717
	400	209,578	47252,438	7,722	60469,828
1000	200	230,585	285260,878	2,081	346271,336
	400	506,998	204702,232	13,262	241308,653
	600	674,958	170182,561	10,090	200424,174
	800	919,434	144295,665	13,010	172724,316

Bảng 2. Kết quả chạy thuật toán GWO

Số lượng điểm cơ sở	*p*	Điều kiện dừng 200 vòng lặp		Điều kiện dừng realtive change < 0,001	
		Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình	Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình
50	10	0,666016	3176,706097	6,227	2627,561
	20	1,365409	2604,834177	11,845	2139,128
	30	2,174421	1936,66822	17,559	1549,124
	40	3,026601	1770,065053	22,933	1405,076
100	20	2,632075	12545,02366	22,527	10143,882
	40	5,896497	8989,045713	44,280	7390,076
	60	9,972773	9666,063428	66,988	5948,535
	80	14,85384	7842,887207	90,684	5486,721
200	40	16,53492	37670,31253	185,695	31590,173
	80	43,52304	40266,06827	261,891	37360,966
	120	67,7809	29076,33119	345,107	25164,859
	160	86,64301	31488,85302	388,953	29634,522
500	100	141,9814	227348,3524	838,411	206241,884
	200	311,6353	191445,156	1252,606	192021,387
	300	616,7325	156397,121	2002,329	160128,311
	400	1036,037	144168,6205	2934,154	150672,738
1000	200	681,1934	768872,0342	2706,252	805187,715
	400	2394,574	598618,1634	6357,754	569069,903
	600	4536,707	504189,6011	10500,002	502642,660
	800	8303,834	449972,4308	17259,767	475587,378

Bảng 3. Kết quả chạy thuật toán BA

Số lượng điểm cơ sở	*p*	Điều kiện dừng 200 vòng lặp		Điều kiện dừng realtive change < 0,001	
		Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình	Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình
50	10	1,475	2568,357	0,466	2802,864
	20	2,704	2048,183	1,278	2222,661
	30	3,922	1499,222	1,037	1615,968
	40	5,077	1252,414	0,848	1468,032
100	20	5,216	8693,561	2,135	9289,357
	40	9,982	5851,829	2,848	6606,554
	60	14,845	5086,598	5,387	5366,386
	80	19,593	3943,945	3,848	4444,389
200	40	31,689	26616,420	11,040	27305,591
	80	60,683	17442,117	16,844	18971,626
	120	78,905	14497,310	12,823	15739,947
	160	83,062	12017,680	11,427	13508,337
500	100	177,173	108386,094	35,618	110908,925
	200	261,557	75080,794	29,521	81469,775
	300	392,035	56096,034	289,717	63754,758
	400	525,391	51379,442	160,839	58259,397
1000	200	573,099	310672,946	197,055	336123,270
	400	1188,759	200220,815	344,984	226760,361
	600	1976,349	178381,611	249,420	190887,008
	800	2332,548	147154,788	397,877	161513,002

Bảng 4. Kết quả chạy thuật toán CSO

Số lượng điểm cơ sở	*p*	Điều kiện dừng 200 vòng lặp		Điều kiện dừng realtive change < 0,001	
		Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình	Thời gian tính toán trung bình(giây)	Giá trị hàm mục tiêu trung bình
50	10	27,200	3198,334	0,405	3221,694
	20	52,302	2580,417	0,653	2528,468
	30	77,057	1851,412	1,241	1972,890
	40	99,908	1733,664	0,997	1712,108
100	20	100,768	10308,869	1,276	10465,188
	40	195,996	7194,263	2,823	7294,443
	60	289,153	5806,017	3,734	5915,410
	80	382,836	4989,526	3,970	5147,218
200	40	600,807	29944,050	7,392	29546,940
	80	1126,286	20588,370	11,671	21132,711
	120	1372,769	17448,491	19,726	17465,152
	160	1553,976	14557,181	16,315	14536,466
500	100	3333,289	117238,816	37,375	117225,533
	200	5018,999	86989,472	52,334	85981,182
	300	7755,437	69246,051	82,997	69410,678
	400	9603,755	61199,911	111,949	61383,185
1000	200	10619,666	345318,585	102,414	348378,638
	400	21472,617	242443,329	235,457	243648,752
	600	31111,272	198407,006	313,125	199612,298
	800	43361,626	172625,576	484,167	172553,015

Dựa trên các kết quả tính toán, ta thấy rằng các thuật toán PSO, GWO, BA, CSO có thể đưa ra một lời giải gần đúng được trong một thời gian hiệu quả. Ta thấy rằng, điều kiện dừng 200 vòng lặp cố định dù làm tăng thời gian xử lí của các thuật toán nhưng bù lại giá trị hàm mục tiêu cho ra tốt hơn so với khi dùng điều kiện dừng $\text{realtive change} < 0{,}001$. Nhìn chung, với điều kiện dừng 200 vòng lặp, thứ tự các thuật toán cho ra kết quả hàm mục tiêu tốt nhất lần lượt là PSO, BA, CSO và GWO. Mặt khác, khi điều kiện dừng $\text{realtive change} < 0{,}001$ được sử dụng thì thứ tự các thuật toán cho ra kết quả hàm mục tiêu tốt nhất lần lượt là BA, CSO, PSO và GWO.

TÀI LIỆU THAM KHẢO

Beck, A., & Sabach, S. (2015). Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, *164*, 1- 40.

Chang, S. C., Yen, W. C. K., Wang, Y. L., & Liu, J. J. (2016). The connected p-median problem on block graphs. *Optimization Letters*, *10*, 1191-1201.

Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat swarm optimization. In *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9* (pp. 854-858). Springer Berlin Heidelberg.

Duy, N. N. Đ., & Hiếu, V. N. M. (2020). Bài toán liên thông p-median trên đồ thị đầy đủ và đồ thị lưỡng phân đầy đủ. *Tạp chí Khoa học Đại học Cần Thơ*, *56*(4), 26-32.

Kennedy, J., & Eberhart, R.(1995) : Particle swarm optimization. *Proc. IEEE International Conf. on Neural Networks (Perth, Australia)*, IEEE Service Center,Piscataway, NJ.

Kien, N. T., Nhan, T., The, W. C. & Hung, N. T. (2021). The Connected *p*-median Problem on Complete Multi-Layered Graphs. *Discrete Mathematics Algorithms and Applications.* 14. 10.1142/S1793830921501184

Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46-61.

X. S. (2010). A new metaheuristic bat- inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Berlin, Heidelberg: Springer Berlin Heidelberg.