



DOI:10.22144/ctujos.2026.118

## THUẬT TOÁN TỐI ƯU HÓA BẦY ĐÀN CHO BÀI TOÁN VỊ TRÍ FERMAT-WEBER TRÊN MẶT CẦU

Ngô Vi Khang<sup>1\*</sup>, Thái Đức Hưng<sup>1</sup>, Phan Văn Hoàng Phát<sup>1</sup>, Hồ Thiện Trung<sup>1</sup> và Nguyễn Thị Cẩm Tiên<sup>2</sup>

<sup>1</sup>Trường Sư phạm, Đại học Cần Thơ, Việt Nam

<sup>2</sup>Trường Khoa học Tự nhiên, Đại học Cần Thơ, Việt Nam

\*Tác giả liên hệ (Corresponding author): Khangb2200016@student.ctu.edu.vn

### Thông tin chung (Article Information)

Nhận bài (Received): 30/09/2025

Sửa bài (Revised): 07/11/2025

Duyệt đăng (Accepted): 17/05/2026

**Title:** Particle Swarm Optimization algorithm for the location problems

**Author(s):** Ngô Vi Khang<sup>1\*</sup>, Thái Đức Hưng<sup>1</sup>, Phan Văn Hoàng Phát<sup>1</sup>, Hồ Thiện Trung<sup>1</sup> and Nguyễn Thị Cẩm Tiên<sup>2</sup>

**Affiliation(s):** <sup>1</sup>School of Education, Can Tho University, Viet Nam; <sup>2</sup>Faculty of Natural Sciences, Can Tho, Viet Nam

### TÓM TẮT

Bài toán Fermat-Weber trên mặt cầu là một mở rộng tự nhiên của bài toán ấy trên mặt phẳng. Do tính phi tuyến và các tính chất hình học đặc biệt của không gian cầu, bài toán đã đặt ra nhiều thách thức trong việc tìm nghiệm tối ưu bằng các phương pháp truyền thống. Trong nghiên cứu này, một cách tiếp cận khác cho bài toán dựa trên thuật toán Tối ưu bầy đàn (Particle Swarm Optimization - PSO) được đề xuất. Thuật toán được thiết kế lại để bảo đảm các cá thể luôn di chuyển trên mặt cầu thông qua kỹ thuật đổi tọa độ.

**Từ khóa:** Bài toán Fermat-Weber trên mặt cầu, bài toán vị trí, bài toán  $p$ -median, tối ưu hóa bầy đàn

### ABSTRACT

The Fermat-Weber problem on a sphere is a natural extension of its planar counterpart. Due to the nonlinearity and unique geometric properties of the spherical space, the problem poses significant challenges in finding optimal solutions using traditional methods. In this study, we propose an alternative approach based on the Particle Swarm Optimization (PSO) algorithm. The algorithm is redesigned to ensure all particles remain constrained to the sphere's surface by employing a coordinate transformation technique.

**Keywords:** Fermat-Weber problem on the Sphere, location problem, Particle Swarm Optimization,  $p$ -median problem

## 1. GIỚI THIỆU

### 1.1. Bài toán Fermat-Weber và các công trình liên quan đến lý thuyết vị trí

Bài toán Fermat-Weber được đặt tên theo Alfred Weber, là một trong những bài toán nổi tiếng nhất trong lý thuyết vị trí. Tiền thân của bài toán là bài toán Fermat được xây dựng bởi nhà toán học Pháp Pierre de Fermat trước năm 1640. Bài toán Fermat được phát biểu như sau: Trong mặt phẳng, hãy xác định một điểm sao cho tổng khoảng cách từ điểm đó

đến ba điểm phân biệt cho trước là nhỏ nhất. Năm 1645, Torricelli đã tìm ra một cách giải bằng hình học cho bài toán ấy, nhưng gần 3 thế kỷ tiếp theo vẫn không có lời giải nào bằng phương pháp số được đưa ra. Cho đến năm 1909, trong cuốn sách *Über den Standort der Industrie*, nhà kinh tế học người Đức Alfred Weber đã tổng quát hóa bài toán Fermat lên thành việc xác định một điểm sao cho tổng khoảng cách có trọng số từ điểm đó đến  $n$  điểm cho trước trên mặt phẳng là nhỏ nhất (Weber, 1909). Bài toán ấy đã đặt nền móng quan trọng cho nhiều công

trình nghiên cứu liên quan đến lý thuyết vị trí (Location Theory). Tác động đầu tiên là việc mô hình hóa các vấn đề trong lý thuyết vị trí: tìm vị trí tối ưu với mục tiêu giảm tổng chi phí vận chuyển hay di chuyển giữa các điểm như nhà máy, bệnh viện, trường học, kho bãi, khu công nghiệp,...; mô hình này cũng được mở rộng bằng việc tính đến trọng số đại diện cho các yếu tố như khối lượng, quy mô hoặc tầm quan trọng của từng điểm. Tiếp đó là sự ra đời của nhiều thuật toán như Thuật toán lặp của Weiszfeld (1937), các thuật toán dựa trên gradient để tìm nghiệm gần đúng như công trình của Kuhn and Kuenne (1962). Đặc biệt, bài toán Fermat-Weber là tiền đề để xây dựng các lớp bài toán  $p$ -median và  $p$ -center ngày nay. Marianov and Eiselt (2024) đã cung cấp một cái nhìn tổng quan có chọn lọc về sự phát triển của lý thuyết vị trí trong suốt 50 năm qua.

### 1.2. Thuật toán Tối ưu bầy đàn

Thuật toán tối ưu bầy đàn (Particle Swarm Optimization hay còn được gọi tắt là PSO) là một metaheuristic nổi tiếng dựa trên quần thể để giải quyết các bài toán tối ưu hóa. Thuật toán này mô phỏng hành vi xã hội của các đàn chim, đàn cá trong quá trình chúng tìm kiếm thức ăn. Chúng liên tục cập nhật vị trí của bản thân theo vị trí tốt nhất mà chúng tìm thấy và vị trí tốt nhất của toàn bộ bầy đàn, dần tập hợp lại và dẫn đến sự hình thành tối ưu. PSO là thuật toán được giới thiệu lần đầu vào năm 1995 bởi nhà tâm lý học xã hội James Kennedy và kỹ sư điện Russell Elberhart (Kennedy & Elberhart, 1995).

Kể từ khi được mọi người biết đến, đã có rất nhiều nghiên cứu và những cải tiến đáng kể đối với thuật toán PSO để phục vụ mục đích và phù hợp với lĩnh vực nghiên cứu của các nhà nghiên cứu. Những sửa đổi PSO có thể kể đến như thuật toán bầy hạt Gauss (Gaussian Swarm) đã được Krohling (2004) phát triển dựa trên phân phối xác suất Gauss (Gaussian probability distribution). Một phiên bản mới của PSO là Concurrent PSO đã được Baskar and Suganthan (2004) nghiên cứu để cải thiện hiệu suất hội tụ của PSO. Các nhà nghiên cứu đã mở rộng phạm vi ứng dụng của thuật toán PSO cho nhiều bài toán tối ưu hóa trên nhiều lĩnh vực, chẳng hạn như tối ưu hóa đa mục tiêu, đa phương thức, tối ưu hóa rời rạc,... Đồng thời, họ cũng tìm hiểu tầm quan trọng của các giá trị tham số trong thuật toán PSO như: trọng số quán tính, hệ số gia tốc, số lượng hạt,... Một số kết quả tiêu biểu: Shi and Eberhart (1998) đã giới thiệu trọng số quán tính ( $w$ ) để kiểm soát sự ảnh hưởng của vận tốc trước đó đến vận tốc

hiện tại của các hạt, giúp giảm nguy cơ xảy ra cực trị cục bộ. Krohling and Coelho (2006) sử dụng các phân phối xác suất như Gauss và Cauchy đề xuất một biến thể mới là Particle Swarm with Exponential Distribution nhằm tăng khả năng tìm kiếm toàn cục và tránh hội tụ sớm.

### 1.3. Tối ưu hóa bầy đàn được áp dụng trong Lý thuyết vị trí

Thuật toán Tối ưu bầy đàn được ứng dụng trong nhiều lĩnh vực khác nhau. Riêng đối với Lý thuyết vị trí, trong những năm gần đây PSO có nhiều ứng dụng quan trọng trong việc giải các bài toán tối ưu liên quan đến các bài toán định vị cơ sở hạ tầng như dịch vụ công (bệnh viện, trường học, trạm cứu hỏa), mạng lưới giao thông và hệ thống logistics. Một cách cụ thể hơn, PSO là một công cụ mạnh để giải các bài toán  $p$ -median và  $p$ -center. Güner and Sevklı (2008) đã phát triển một thuật toán PSO rời rạc nhằm giải bài toán định vị cơ sở không giới hạn công suất (Uncapacitated Facility Location Problem-UFLP), với mục tiêu tối thiểu hóa tổng chi phí thiết lập và vận chuyển. Sevklı et al. (2014) đã xây dựng một biến thể của PSO là New Discrete Particle Swarm Optimization nhằm giải quyết bài toán  $p$ -median, với mục tiêu giảm chi phí phân bổ và cải thiện chất lượng nghiệm. Gần đây nhất, Zheng et al. (2024) đã đề xuất thuật toán PSO cải tiến mang tên Diversity-based Hybrid PSO để giải bài toán định vị trung tâm logistics cho drone, tận dụng các chiến lược tìm kiếm đa dạng nhằm nâng cao khả năng thoát khỏi cực tiểu cục bộ và cải thiện hiệu quả định vị.

### 1.4. Các đóng góp của bài báo

Trong bài báo này, thuật toán Tối ưu bầy đàn được ứng dụng vào việc tìm nghiệm xấp xỉ cho bài toán Fermat-Weber trên mặt cầu.

## 2. BÀI TOÁN FERMAT-WEBER TRÊN MẶT CẦU

### 2.1. Giới thiệu

Trong bài báo này, bài toán Fermat-Weber đã được mở rộng để sử dụng lên trên mặt cầu. Không gian nghiên cứu chuyển từ không gian Euclid sang không gian phi Euclid, cụ thể là không gian hình học cầu, nơi khoảng cách giữa hai điểm trên mặt cầu được đo bằng độ dài cung tròn lớn thay vì đoạn thẳng nối hai điểm đó. Bài toán Fermat-Weber trên mặt cầu đã được nghiên cứu bởi nhiều tác giả trước đó, tiêu biểu như Katz and Cooper (1980) đã mô hình hóa khoảng cách giữa các điểm bằng độ dài cung tròn lớn và xây dựng điều kiện cân bằng vector

để xác định nghiệm tối ưu, đồng thời đề xuất một thuật toán lặp tương tự như Weiszfeld để tìm nghiệm gần đúng cho bài toán, hay Rubio-López et al. (2023) xem xét bài toán Weber ngược trên mặt phẳng và mặt cầu để tìm các trọng số khi biết trước nghiệm tối ưu của bài toán.

Trên mặt cầu, bài toán Fermat-Weber yêu cầu tìm một điểm  $P$  sao cho tổng khoảng cách từ điểm đó đến các điểm  $A_1, A_2, \dots, A_n$  đạt giá trị nhỏ nhất. Đối với bài toán này, khoảng cách giữa 2 điểm cùng nằm trên một mặt cầu cho trước được tính dựa trên độ dài cung tròn lớn giữa các điểm thay vì khoảng cách thẳng. Bài toán được mô tả bởi phương trình tối ưu sau:

$$S(P) = \min_{p \in S} \sum_{i=1}^n w_i \cdot d(P, A_i).$$

Trong đó:

- $S(P)$  là khoảng cách tối ưu cần tìm;
- $w_i$  là trọng số của điểm  $A_i$ ;
- $d(P, A_i)$  là khoảng cách giữa hai điểm trên mặt cầu.

Việc nghiên cứu bài toán Fermat-Weber trên mặt cầu không chỉ mang tính lý thuyết mà còn đem lại nhiều ứng dụng thực tế. Điển hình, bài toán này có thể được áp dụng trong tối ưu hóa vị trí cơ sở hạ tầng như trạm phát sóng, cảng biển và sân bay trên bề mặt Trái Đất nhằm giảm thiểu chi phí vận hành nhưng vẫn đạt hiệu suất hoạt động tốt nhất. Ngoài ra, bài toán còn đóng vai trò quan trọng trong hệ thống định vị vệ tinh (Global Positioning System-GPS), lập kế hoạch đường bay ngắn nhất trong hàng không. Những vấn đề này đã được nghiên cứu rộng rãi trong lĩnh vực facility location problem với các nghiên cứu về vị trí cơ sở trên mặt cầu như Katz and Cooper (1980), Drezner and Wesolowsky (1978) và gần đây là Shih (2015). Những ứng dụng này khẳng định tầm quan trọng của bài toán trong việc giải quyết các vấn đề thực tiễn liên quan đến định vị và tối ưu hóa trên bề mặt cầu. Bên cạnh các hướng tiếp cận dựa trên thuật toán lặp như các thuật toán Weiszfeld mở rộng được mô tả bởi Drezner and Wesolowsky (1978) và Katz and Cooper (1980), phương pháp Big Triangle–Small Triangle được đề xuất bởi Suzuki (2019) nhằm giải bài toán Weber trên mặt cầu thông qua cách chia miền tìm kiếm thành các tam giác cầu lớn, sau đó dần thu hẹp thành các tam giác nhỏ hơn để xác định nghiệm gần tối ưu. Ở mỗi bước lặp, miền tìm kiếm được tinh chỉnh dựa trên giá trị của hàm mục tiêu tại các đỉnh tam giác, từ đó dần hội tụ đến vị trí tối ưu trên mặt cầu. Cách

tiếp cận này đặc biệt hữu ích trong việc xử lý các bài toán phi lồi trên bề mặt cầu và khắc phục các hạn chế về hội tụ của những thuật toán dạng Weiszfeld cổ điển (Suzuki, 2019). Nguyen et al. (2024) đã đề xuất một biến thể của thuật toán PSO, gọi là Fermat–Weber Location PSO (FWL-PSO), nhằm giải quyết bài toán lập kế hoạch đường bay hợp tác của các thiết bị bay không người lái (UAVs). Trong nghiên cứu này, bài toán Fermat–Weber được phát biểu trong không gian Euclid  $\mathbb{R}^n$ , nơi khoảng cách giữa các điểm được tính bằng khoảng cách Euclid thông thường và mục tiêu là tối ưu vị trí tập trung của các UAV trong không gian ba chiều. Tuy nhiên, cách tiếp cận này chưa xem xét bài toán Fermat–Weber trong không gian phi Euclid, cụ thể là trên mặt cầu, nơi khoảng cách được đo theo độ dài cung tròn lớn. Hiện nay, chưa có công trình nào áp dụng trực tiếp thuật toán PSO để giải bài toán Fermat–Weber trên mặt cầu, do đó hướng tiếp cận của nghiên cứu này góp phần mở rộng phạm vi ứng dụng của PSO sang không gian hình học cầu.

## 2.2. Một số tính chất

Ta xây dựng hàm mục tiêu của bài toán trên mặt cầu đơn vị:

$$S^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}.$$

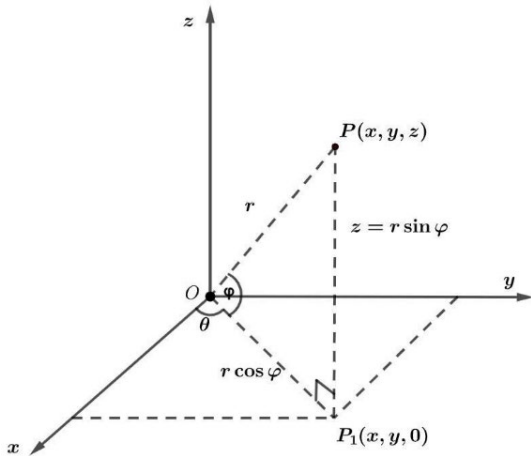
Cho  $n$  điểm phân biệt,  $p_i \in S^2, i = 1, 2, \dots, n$ , và các trọng số  $w_i > 0, \forall i = 1, 2, \dots, n$ , bài toán Fermat-Weber trên mặt cầu yêu cầu tìm điểm  $p_0 \in S^2$  để tối thiểu hóa hàm Weber:

$$F(p) = \sum_{i=1}^n w_i d_{S^2}(p, p_i).$$

### Khái niệm 1: Hệ tọa độ cầu

Trong không gian  $Oxyz$ , tọa độ cầu của một điểm  $P(x; y; z)$  là một bộ có thứ tự gồm 3 thành phần  $(r; \varphi; \theta)$  trong đó:  $r = OP$ ,  $\theta$  là góc giữa vector  $\vec{i} = (1; 0; 0)$  và  $\overline{OP_1}$ ,  $\varphi$  là góc giữa vector  $OP$  và mặt phẳng  $(Oxy)$  với  $P_1$  là hình chiếu của  $P$  lên mặt phẳng  $(Oxy)$ ,  $\varphi \in \left[ \frac{-\pi}{2}, \frac{\pi}{2} \right]$  và  $\theta \in (-\pi, \pi]$ .

Kí hiệu:  $P(r; \varphi; \theta)$ .



**Hình 1. Hệ tọa độ cầu**

Để giải bài toán Fermat-Weber trên mặt cầu với các điểm nguồn (đại diện cho các cơ sở, thành phố, khu vực,...), cần thiết phải định nghĩa một hệ tọa độ cầu liên kết trực tiếp với tọa độ địa lý của các điểm đó. Do đó ta quy ước:

- Góc  $\varphi$  là vĩ độ địa lý và:
  - +  $\varphi > 0$ : điểm nằm ở Bắc bán cầu,
  - +  $\varphi < 0$ : điểm nằm ở Nam bán cầu,
  - +  $\varphi = 0$ : điểm nằm trên mặt phẳng xích đạo

(Oxy).

- Góc  $\theta$  là kinh độ địa lý và:
  - +  $\theta > 0$ : điểm nằm ở bán cầu Đông,
  - +  $\theta < 0$ : điểm nằm ở bán cầu Tây,
  - +  $\theta = 0$ : điểm nằm trên kinh tuyến gốc.

Từ khái niệm ta có: 
$$\begin{cases} x = r \cos \varphi \cos \theta \\ y = r \cos \varphi \sin \theta \\ z = r \sin \varphi \end{cases}$$

Trong hệ tọa độ này, khi  $r = 1$  tọa độ một điểm  $P(r, \varphi, \theta)$  được kí hiệu là:  $P(\varphi; \theta)$ .

**Định lý 1:** Trong không gian Oxyz, một điểm  $P(x; y; z)$  trên mặt cầu đơn vị  $S^2$  có tọa độ trong hệ tọa độ cầu là  $P(\varphi; \theta)$  với  $\varphi = \arcsin z$  và

$$\theta = \begin{cases} \arctan \frac{y}{x}, x > 0, y > 0 \\ \arctan \frac{y}{x} + \pi, x < 0, y > 0 \\ \arctan \frac{y}{x} - \pi, x < 0, y < 0 \\ \arctan \frac{y}{x}, x > 0, y < 0 \\ \frac{\pi}{2}, x = 0, y > 0 \\ \frac{-\pi}{2}, x = 0, y < 0 \end{cases}$$

Trong nghiên cứu này, dữ liệu đầu vào là tọa độ của các điểm nguồn trên mặt cầu được biểu thị với đơn vị độ. Tuy nhiên, khi thực hiện tính toán vị trí và khoảng cách giữa các điểm trên mặt cầu cần sử dụng các hàm lượng giác như  $\sin(\ )$ ,  $\cos(\ )$  và  $\arccos(\ )$  trong Python mặc định sử dụng đơn vị radian. Do đó, định lý này là nền tảng để xây dựng các định lý tiếp theo và để đảm bảo tính đúng đắn cho các tính toán trong bài báo.

**Định lý 2:** Độ dài cung ngắn nhất giữa hai điểm  $P(R; \varphi; \theta)$  và  $A_i(R; \varphi_i; \theta_i)$  trên mặt cầu có bán kính R được xác định như sau:

$$d(P, A_i) = R \arccos[\cos \varphi \cos \varphi_i \cos(\theta - \theta_i) + \sin \varphi \sin \varphi_i].$$

**Chứng minh:**

Xét trên hệ trục ta xác định được tọa độ các điểm như sau:

$$P(R \cos \varphi \cos \theta; R \cos \varphi \sin \theta; R \sin \varphi);$$

$$A_i(R \cos \varphi_i \cos \theta_i; R \cos \varphi_i \sin \theta_i; R \sin \varphi_i).$$

Xét mặt phẳng cắt mặt cầu tại đường tròn có O là tâm chứa P và  $A_i$ .

Khi đó ta được:

$$\overline{OP} \cdot \overline{OA_i} = |\overline{OP}| \cdot |\overline{OA_i}| \cdot \cos(\widehat{POA_i}).$$

$$\Rightarrow R^2 \cos(\widehat{POA_i}) = R^2 (\cos \varphi \cos \theta \cdot \cos \varphi_i \cos \theta_i + \cos \varphi \sin \theta \cdot \cos \varphi_i \sin \theta_i + \sin \varphi \sin \varphi_i).$$

$$\Rightarrow \cos(\widehat{POA_i}) = \cos \varphi \cos \theta \cdot \cos \varphi_i \cos \theta_i + \cos \varphi \sin \theta \cdot \cos \varphi_i \sin \theta_i + \sin \varphi \sin \varphi_i.$$

$$\begin{aligned} \Rightarrow \cos(\widehat{POA_i}) &= \cos \varphi \cos \varphi_i \cos(\theta - \theta_i) + \\ &+ \sin \varphi \sin \varphi_i. \\ \Rightarrow \widehat{POA_i} &= \arccos[\cos \varphi \cos \varphi_i \cos(\theta - \theta_i) + \\ &+ \sin \varphi \sin \varphi_i]. \end{aligned}$$

Khi đó độ dài cung cần tìm là:

$$\begin{aligned} d(P, A_i) &= R \cdot \widehat{POA_i} \\ &= R \arccos(\cos \varphi \cos \varphi_i \cos(\theta - \theta_i) + \sin \varphi \sin \varphi_i). \end{aligned}$$

Như vậy, định lý đã được chứng minh.

**Định lý 3:** Cho 2 điểm  $X_1 = X(\phi_1, \theta_1)$ ,  $X_2 = X(\phi_2, \theta_2) \in S^2$ , khi đó độ dài cung ngắn nhất  $\alpha = \alpha(X_1, X_2)$  thoả mãn:

$$\sin\left(\frac{\alpha}{2}\right) = \frac{1}{2} \sqrt{2 - 2 \sin \phi_1 \sin \phi_2 \cos(\theta_1 - \theta_2)}.$$

**Chứng minh:** đổi sang hệ tọa độ cầu ta có:

$$\begin{aligned} X_1 &= X(\phi_1, \theta_1) = (\cos \phi_1 \cos \theta_1, \cos \phi_1 \sin \theta_1, \sin \phi_1); \\ X_2 &= X(\phi_2, \theta_2) = (\cos \phi_2 \cos \theta_2, \cos \phi_2 \sin \theta_2, \sin \phi_2). \end{aligned}$$

Cắt hình cầu đơn vị theo mặt phẳng  $(OX_1X_2)$  ta được hình tròn lớn như hình 2, ta có:

$$\|MX_2\| = \frac{\|X_1 - X_2\|}{2}.$$

Khi đó:

$$\sin\left(\frac{\alpha}{2}\right) = \frac{\|X_1 - X_2\|}{2}.$$

Ta lại có:

$$\begin{aligned} \|X_1 - X_2\|^2 &= \|X(\phi_1, \theta_1) - X(\phi_2, \theta_2)\|^2 \\ &= 2 - 2 \sin \phi_1 \sin \phi_2 \cos(\theta_1 - \theta_2). \end{aligned}$$

Vì vậy:

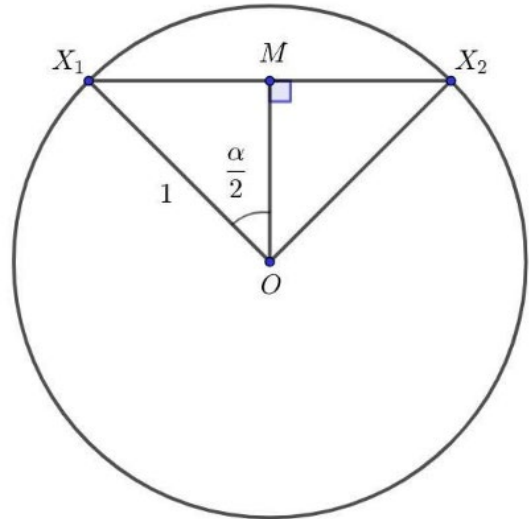
$$\sin\left(\frac{\alpha}{2}\right) = \frac{1}{2} \sqrt{2 - 2 \sin \phi_1 \sin \phi_2 \cos(\theta_1 - \theta_2)}. \quad \square$$

**Định nghĩa 1: Điểm đối cực của một điểm trên mặt cầu** là điểm nằm đối xứng với nó qua tâm của mặt cầu. Điểm đối cực của  $r(\varphi, \theta)$  là  $\bar{r}(-\varphi, \theta + \pi)$ .

Định lý 2 và định lý 3 là nền tảng để xây dựng hàm mục tiêu cho bài toán Fermat-Weber. Trong đó, định lý 2 giúp xác định công thức tính khoảng cách trắc địa (geodesic distance) giữa hai điểm trên mặt

cầu bán kính  $R$ , định lý 3 giúp xác định miền khả thi và tính duy nhất của cung ngắn nhất nối hai điểm không đối cực trên mặt cầu, nhờ đó giúp hàm mục tiêu được xác định tốt.

Trong trường hợp tồn tại điểm nguồn  $A_i$  có trọng số  $w_i < 0$ , ta thay  $A_i$  bằng điểm đối cực của nó là  $A_i'$  với trọng số  $-w_i > 0$  và tiếp tục xây dựng hàm mục tiêu với các trọng số dương để giải quyết bài toán.



Hình 2. Hình tròn lớn

### 3. THUẬT TOÁN TỐI ƯU BẦY ĐÀN (PSO)

#### 3.1. Mô tả thuật toán

Trong mô hình này, các cá thể trong bầy được xem như các hạt (particle) di chuyển trong không gian tìm kiếm và mỗi hạt đại diện cho một giải pháp tiềm năng cho bài toán đã cho. Kennedy and Eberhart (1995) gọi  $P_{best}$  (Personal best value) và  $G_{best}$  (Global best value) là hai giá trị cơ bản. Trong đó,  $P_{best}$  là vị trí tốt nhất mà mỗi cá thể tìm thấy được và  $G_{best}$  là vị trí tốt nhất mà toàn bộ bầy đàn thu nhận được trong không gian tìm kiếm.

Ngoài ra, các đại lượng chính trong PSO được định nghĩa và kí hiệu như sau:

**Vị trí (Position):** Vị trí của hạt thứ  $i$  trong không gian tìm kiếm tại thời điểm  $t$ , ký hiệu là  $x_i^{(t)}$ .

**Vận tốc (Velocity):** Vận tốc của hạt  $i$  tại thời điểm  $t$  ký hiệu là  $v_i^{(t)}$ , quyết định hướng và khoảng cách di chuyển của hạt trong mỗi lần lặp.

**Kinh nghiệm cá nhân** ( $P_{i,best}^k$ ): Vị trí tốt nhất mà mỗi hạt đạt được dựa trên giá trị hàm mục tiêu.

**Kinh nghiệm toàn cục** ( $G_{best}^k$ ): Vị trí tốt nhất của cả bầy đàn đạt được trong quá trình tìm kiếm.

**3.2. Mã giả của thuật toán**

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

**Bước 1: Khởi tạo các tham số và quần thể hạt**

Khởi tạo các hằng số  $c_1, c_2, k_{max}, r_1, r_2$ .

Khởi tạo ngẫu nhiên  $n$  vị trí và vận tốc ban đầu của các hạt trong không gian tìm kiếm.

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

Trong đó:

$w$ : Hệ số quán tính, giúp kiểm soát mức độ ảnh hưởng của vận tốc cũ.

$c_1, c_2$ : Hệ số học hỏi thể hiện sự quan tâm của hạt đến kinh nghiệm cá nhân và toàn cục.

$r_1, r_2$ : Các số ngẫu nhiên trong đoạn  $[0,1]$ .

Tính toán giá trị hàm mục tiêu tại vị trí của từng hạt.

Đặt giá trị ban đầu cho  $P_{i,best}^1$  là vị trí hiện tại của hạt  $i$ .

**Bước 2: Vòng lặp cập nhật quần thể**

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

**While**  $k \leq k_{max}$

+ **Cập nhật vận tốc và vị trí của cá thể:**

Vận tốc của mỗi hạt được cập nhật theo công thức:

$$v_i^{(t+1)} = w \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (P_{i,best}^k - x_i^{(t)}) + c_2 \cdot r_2 \cdot (G_{best}^k - x_i^{(t)}).$$

Công thức này được xây dựng trên cơ chế điều chỉnh vị trí và vận tốc của mỗi hạt dựa trên thông tin thu được từ kinh nghiệm cá nhân và kinh nghiệm toàn cục, để mỗi hạt di chuyển dần về các vùng có giá trị mục tiêu tốt hơn.

Công thức vận tốc mới  $v_i^{(t+1)}$  được tạo thành từ tổng của ba thành phần chính:

- **Thành phần quán tính:**  $w \cdot v_i^t$

Thành phần này có chức năng khám phá các vùng giải pháp mới trên bản đồ, làm cho các hạt di chuyển cùng hướng và cùng vận tốc theo hướng và vận tốc cũ trước đó.

Nếu  $w = 1$  thì chuyển động của hạt hoàn toàn bị ảnh hưởng bởi chuyển động trước đó. Vì vậy, hạt có thể tiếp tục đi theo cùng một hướng.

Nếu  $0 < w < 1$  thì ảnh hưởng của chuyển động trước đó bị giảm đi, nghĩa là hạt sẽ chuyển sang các vùng khác trong không gian tìm kiếm.

Khi hệ số quán tính  $w$  càng cao thì càng giúp các hạt bay xa hơn theo hướng cũ, bay vào những vùng chưa được khám phá và ít bị ảnh hưởng bởi các giải pháp tốt đã biết. Lúc đầu, thuật toán thường cần  $w$  cao để khám phá toàn bộ không gian. Càng về sau,  $w$  thường được giảm dần để thuật toán khai thác và hội tụ về giải pháp tốt nhất mà nó tìm được.

• **Thành phần ảnh hưởng cá nhân:**

$$c_1 \cdot r_1 \cdot (P_{i,best}^k - x_i^{(t)})$$

Thành phần này giúp cho hạt quay lại vị trí trước đó, tốt hơn vị trí hiện tại. Nếu vị trí hiện tại của hạt ( $x_i^{(t)}$ ) càng xa vị trí tối ưu mà hạt đã tìm được ( $P_{i,best}^k$ ) thì thành phần ( $P_{i,best}^k - x_i^{(t)}$ ) càng lớn, nghĩa là điều này làm cho các hạt có xu hướng quay lại vị trí tốt nhất trước đó của nó, tốt hơn vị trí hiện tại và đại diện cho xu hướng tự cải thiện cá nhân. Tham số  $c_1$  là hệ số học hỏi, đánh giá tầm quan trọng của những kinh nghiệm trong quá khứ của chính hạt đó. Tham số ngẫu nhiên  $r_1$  đóng vai trò thiết yếu trong việc tránh hội tụ sớm, làm tăng khả năng đạt được tối ưu toàn cục.

• **Thành phần ảnh hưởng xã hội:**

$$c_2 \cdot r_2 \cdot (G_{best}^k - x_i^{(t)})$$

Thành phần này đại diện cho sự học hỏi hay bắt chước những cá thể thành công nhất trong bầy đàn. Độ chênh lệch ( $G_{best}^k - x_i^{(t)}$ ) hoạt động như một lực hút các hạt về điểm tốt nhất được tìm thấy cho đến vòng lặp  $t$ . Tương tự, tham số  $c_2$  là hệ số học hỏi xã hội và nó đánh giá tầm quan trọng của việc học tập tối ưu toàn cục của bầy đàn. Tham số ngẫu nhiên  $r_2$  cũng đóng vai trò hoàn toàn giống như  $r_1$ .

Hai thành phần ảnh hưởng cá nhân và ảnh hưởng xã hội có chức năng tìm kiếm kỹ lưỡng các nghiệm tối ưu xung quanh một khu vực đã biết gần ( $P_{i,best}^k$  và  $G_{best}^k$ ) để tìm ra giải pháp tốt nhất có thể trong vùng đó.

Vị trí của hạt được cập nhật bằng cách cộng vận tốc vào vị trí hiện tại:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}.$$

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

**If**  $f(x_i^k) > f(P_{i,best}^k)$  **then**  $P_{i,best}^k := P_{i,best}^{k-1}$ ,

**Else**  $P_{i,best}^k := x_i^k$

**End If**

+ **Cập nhật vị trí tối ưu của cả bầy đàn:**

$$G_{best}^k = \arg \min \{ f(P_{i,best}^k) \}.$$

Tiếp tục gán  $k := k + 1$

**Endwhile**

**Output:** Vị trí tối ưu  $G_{best}^j$  và giá trị tối ưu  $f(G_{best}^j)$ .

#### 4. ỨNG DỤNG CỦA THUẬT TOÁN TỐI ƯU BẦY ĐÀN CHO BÀI TOÁN FERMAT-WEBER TRÊN MẶT CẦU

##### 4.1. Mô tả các tính toán

**Thuật toán 1: Đổi tọa độ**

**Input:** Đổi tọa độ từ độ sang radian

Cho một danh sách các tọa độ của các thành phố trên bản đồ dưới dạng đơn vị độ: tọa độ đầu là của kinh tuyến, tọa độ sau là của vĩ tuyến và các trọng số tương ứng.

*Ví dụ:* Tọa độ của thủ đô Hà Nội trên bản đồ là  $21,02450^\circ$  vĩ độ Bắc và  $105,84117^\circ$  kinh độ Đông, khi đó mỗi thành phố sẽ tương ứng một bộ số  $(21,0245 \text{ B}; 105,84117 \text{ D}; w)$ , trong đó  $w$  là trọng số tương ứng.

**Output:** Cho ra lại một danh sách các thành phố trên thành phố dưới dạng radian.

Ta xử lý vĩ tuyến: Nếu chữ cái cuối của tọa độ là T (Tây) thì đổi dấu, còn nếu chữ cái cuối là D (Đông) thì giữ nguyên.

Công thức đổi tọa độ từ một điểm  $(\varphi, \theta)$  từ độ sang radian:

$$c = \frac{\varphi \cdot \pi}{180} \qquad d = \frac{\theta \cdot \pi}{180}$$

Với  $c, d$  là các thành phần tọa độ của thành phố sau khi đã đổi sang radian.

##### Thuật toán 2: Xây dựng hàm mục tiêu

**Input:** Cho danh sách tọa độ của các thành phố

Cập nhật các hằng số quán tính  $w$  của các thành phố như trong Bảng 1.

Trong nghiên cứu này, Trái Đất được chuẩn hóa như một hình cầu bán kính  $R$ . Do đó theo định lý 3, khoảng cách giữa hai điểm trên Trái Đất chính là khoảng cách ngắn nhất giữa hai điểm  $P(R; \varphi; \theta)$  và  $A_i(R; \varphi_i; \theta_i)$  trên mặt cầu bán kính  $R = 6371$  (ta chọn  $R$  bằng bán kính trung bình của Trái Đất):

$$d(P, A_i) = 6371 \cdot \arccos[\cos \varphi \cos \varphi_i \cos(\theta - \theta_i) + \sin \varphi \sin \varphi_i].$$

Trong đó:  $\varphi, \varphi_i \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right]$  và  $\theta, \theta_i \in (-\pi, \pi]$ .

Ở đây  $\varphi, \varphi_i$  là vĩ độ và  $\theta, \theta_i$  là kinh độ của các điểm tương ứng.

**Output:** Hàm tối ưu là:

$$f(X) = \sum_{i=1}^n w_i d(X, A_i).$$

##### 4.2. Cấu hình máy dữ liệu

**Ta xét một bài toán mô hình hóa về việc tìm điểm tối ưu trên mặt cầu:**

**Bài toán:** Một công ty logistics toàn cầu muốn mở rộng quy mô hoạt động bằng cách xây dựng một trung tâm điều phối vận hành để tối ưu hóa thời gian và chi phí vận chuyển hàng hóa từ trung tâm này đến các thành phố nơi công ty mẹ có các công ty con trực thuộc.

Bài toán đang xét trên quy mô 30 quốc gia/ vùng lãnh thổ, mỗi quốc gia được xác định bởi 1 bộ tọa độ (kinh độ, vĩ độ) và trọng số  $w_i$  (phản ánh mức độ quan trọng của thành phố  $i$ , được xác định dựa trên dân số, số đơn hàng, quy mô kinh tế của thành phố đó, trong mô hình này trọng số chỉ mang tính giả định không phản ánh thực tế), được thể hiện trong Bảng 1.

**Bảng 1. Danh sách các thành phố**

STT	Tên quốc gia	Tên thành phố	Kinh độ	Vĩ độ	Trọng số $w_i$
1	Mỹ	New York	74,00597 T	40,71427 B	10,0
2	Việt Nam	Hà Nội	105,84117 D	21,02450 B	5,8
3	Anh	London	0,12574 T	51,50853 B	9,5
4	Pháp	Paris	2,34880 D	48,85341 B	9,2
5	Nhật Bản	Tokyo	139,69171 D	35,68950 B	10,0
6	Trung Quốc	Bắc Kinh	116,39723 D	39,90750 B	9,8
7	Úc	Sydney	151,20732 D	33,86785 N	8,5
8	Nam Phi	Cape Town	18,42322 D	33,92584 N	3,5
9	Brazil	Rio de Janeiro	43,18223 T	22,90642 N	7,0
10	Nga	Moscow	37,61556 D	55,75222 B	8,8
11	Ấn Độ	Mumbai	72,89261 D	19,07283 B	9,4
12	Canada	Toronto	79,39864 T	43,70643 B	8,5
13	Mexico	Mexico City	99,12766 T	19,42847 B	9,6
14	Ai Cập	Cairo	31,24967 D	30,06263 B	6,7
15	UAE	Dubai	55,30927 D	25,07725 B	9,0
16	Argentina	Buenos Aires	58,37723 T	34,61315 N	7,2
17	Đức	Berlin	13,41053 D	52,52437 B	8,6
18	Ý	Rome	12,51133 D	41,89193 B	8,0
19	Thái Lan	Bangkok	100,50144 D	13,75398 B	7,8
20	Hàn Quốc	Seoul	126,97840 D	37,56600 B	9,1
21	Nigeria	Lagos	3,39467 D	6,45407 B	6,5
22	Kenya	Nairobi	36,81667 D	1,28333 N	4,0
23	Indonesia	Jakarta	106,84513 D	6,21462 N	7,5
24	Tây Ban Nha	Madrid	3,70256 T	40,41650 B	8,4
25	Singapore	Singapore	103,85007 D	1,28967 B	9,5
26	Peru	Lima	77,02824 T	12,04318 N	5,4
27	Hy Lạp	Athens	23,72784 D	37,98376 B	7,0
28	Thổ Nhĩ Kỳ	Istanbul	28,94966 D	41,01384 B	8,1
29	Ba Lan	Warsaw	21,01178 D	52,22977 B	7,6
30	Ả Rập Xê Út	Riyadh	46,72185 D	24,68773 B	8,1

Ghi chú: Tọa độ các thành phố được trích xuất từ cơ sở dữ liệu của GeoNames (<https://www.geonames.org/>). Mã nguồn thực hiện quá trình thu thập và xử lý dữ liệu có thể được truy cập tại liên kết sau: [https://colab.research.google.com/drive/1E11yu-XDmYVxfywHYyie0i\\_7Uu5J4x9C?usp=sharing](https://colab.research.google.com/drive/1E11yu-XDmYVxfywHYyie0i_7Uu5J4x9C?usp=sharing)

Bài toán được xem xét và giải bằng thuật toán Tối ưu bầy đàn được tiến hành trên một máy tính cá nhân với cấu hình như sau: Hệ điều hành Windows 11 Home Single Language 64-bit (10,0, Build 26100), bộ xử lý (CPU) Core™ i5-11320H, 3.20 GHz (8 CPU luồng) – 2.5GHz, bộ nhớ RAM 8GB, ổ cứng SSD 512 GB.

Việc chạy thuật toán và vẽ bản đồ bằng ngôn ngữ lập trình Python được thực hiện trên môi trường lập trình Google Colab.

Toàn bộ mã nguồn phục vụ cho việc giải bài toán được công bố công khai tại:

<https://colab.research.google.com/drive/1--b1sFWn1DJGj7PEqrz1QaEcU3efpaQn?usp=sharing>

**4.3. Kết quả chạy thuật toán**



**Hình 3. Vị trí các thành phố lớn trên thế giới**

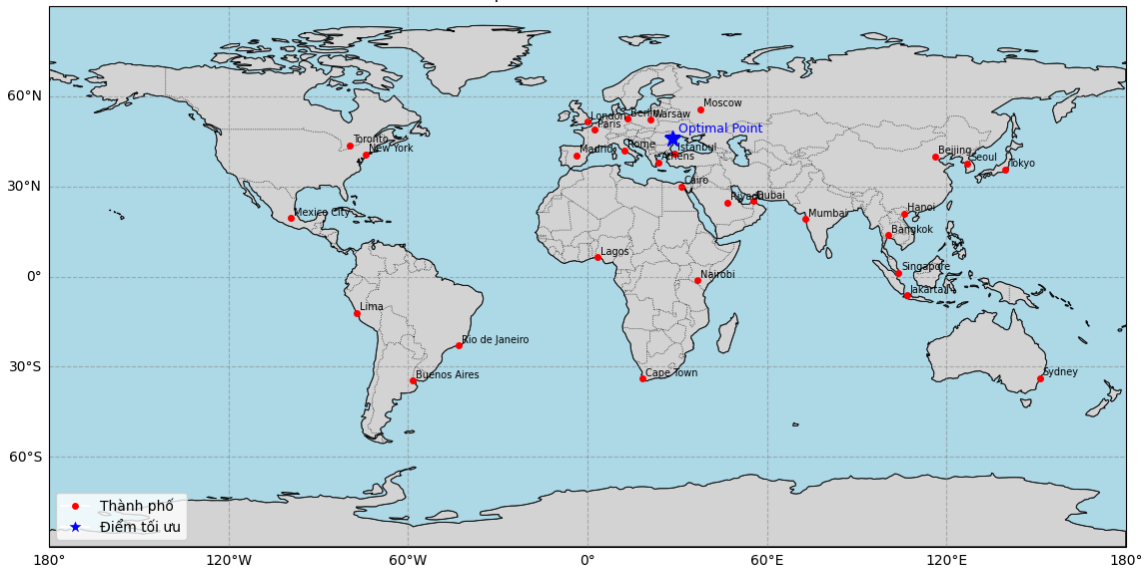
Dựa trên các kết quả tính toán, điều chỉnh số lượng điểm cơ sở (điểm vị trí), số hạt PSO và số vòng lặp nhằm khảo sát thời gian xử lý của thuật

toán và các sai số, các kết quả cụ thể được cho dưới bảng sau:

**Bảng 2. Kết quả chạy thuật toán PSO**

Số vòng lặp	Số hạt	Thời gian (giây)	Kết quả tối ưu	Điểm tối ưu (Kinh độ, vĩ độ)
50	10	0,02449656	1396071,49	28,2541 D, 46,0562 B
	30	0,07775974	1396071,93	28,2252 D, 46,0221 B
	50	0,12940884	1396071,50	28,2639 D, 46,0624 B
100	10	0,05893683	1396071,49	28,2607 D, 46,0544 B
	30	0,20696425	1396071,49	28,2608 D, 46,0543 B
	50	0,38108444	1396071,49	28,2608 D, 46,0544 B
300	10	0,14371610	1396071,49	28,2607 D, 46,0543 B
	30	0,44100761	1396071,49	28,2607 D, 46,0543 B
	50	1,24601293	1396071,49	28,2607 D, 46,0543 B

Các thành phố và điểm Fermat-Weber tối ưu



**Hình 4. Vị trí các thành phố và điểm tối ưu trên bản đồ thế giới**

**4.4. So sánh thuật toán PSO và thuật toán Weiszfeld mở rộng**

Ở phần này, mức độ tối ưu của thuật toán PSO được so với một thuật toán tối ưu khác đã được sử dụng để giải quyết bài toán Fermat-Weber trên mặt cầu là thuật toán Weiszfeld mở rộng trên mặt cầu được xây dựng bởi Katz and Cooper (1980).

Dưới đây là kết quả thu được khi so sánh thời gian và độ chính xác giữa hai thuật toán PSO và Weiszfeld khi chạy hai thuật toán với cùng bộ  $n$  điểm nguồn được sinh ngẫu nhiên trên bề mặt cầu, đồng thời mỗi điểm đều được gán trọng số ngẫu nhiên trên đoạn  $[1; 10]$ :

**Bảng 3.  $n = 200$**

Thuật toán	PSO	Weiszfeld mở rộng
<b>Kết quả</b>		
Thời gian ( giây )	0,2200	0,1160
Điểm tối ưu	165,3463 T, 5,6654 B	165,5550 T, 5,4881 B
Tổng khoảng cách tối ưu	10202803,65	10202442,79

**Bảng 4.**  $n = 500$

Thuật toán	PSO	Weiszfeld mở rộng
<b>Kết quả</b>		
Thời gian (giây)	0,4868	0,2773
Điểm tối ưu	162,1655 D, 5,1038 N	161,4155 D, 4,6684 N
Tổng khoảng cách tối ưu	27317356,92	27317260,56

**Bảng 5.**  $n = 1000$

Thuật toán	PSO	Weiszfeld mở rộng
<b>Kết quả</b>		
Thời gian (giây)	1,0539	0,6172
Điểm tối ưu	134,4171 D, 14,1236 B	134,4662 D, 14,0449 B
Tổng khoảng cách tối ưu	54405129,01	54405126,08

Toàn bộ mã nguồn phục vụ cho việc so sánh thời gian chạy và độ chính xác của hai thuật toán được công bố công khai tại: [https://colab.research.google.com/drive/1bevY6ZK0B5vY\\_tz4CnEGFEjwjmOoCjmm?usp=sharing](https://colab.research.google.com/drive/1bevY6ZK0B5vY_tz4CnEGFEjwjmOoCjmm?usp=sharing)

Qua ba quy mô thực nghiệm, ta thấy hai thuật toán này cho ra kết quả tối ưu và điểm tối ưu mặc dù có mức sai lệch nhưng không đáng kể, mức sai lệch có thể giảm xuống khi ta tăng số hạt và số vòng lặp của thuật toán PSO lên. Về mặt thời gian, ta có thể thấy thuật toán Weiszfeld có tốc độ xử lý nhanh hơn thuật toán PSO, lý do là ở mỗi vòng lặp, thuật toán Weiszfeld chỉ cần tính toán và cập nhật, nhưng thuật toán PSO thì cần đánh giá cả một quần thể, mặc dù vậy đây vẫn là một hướng tiếp cận mới đối với bài toán này, hứa hẹn có thể phát triển và mở rộng trong tương lai.

### 5. KẾT QUẢ VÀ THẢO LUẬN

Trong bài báo này, thuật toán Tối ưu bầy đàn (PSO) đã được ứng dụng để tìm ra nghiệm xấp xỉ cho bài toán Fermat-Weber trên mặt cầu. Kết quả cho thấy thuật toán có khả năng hội tụ đến nghiệm xấp xỉ với sai số nhỏ, đồng thời đảm bảo tính ổn định qua nhiều lần chạy độc lập. Việc triển khai PSO trên mặt cầu cũng cho thấy tính linh hoạt của thuật toán khi mở rộng sang các dạng hình học khác ngoài không gian phẳng. Kết quả nghiên cứu bước đầu khẳng định tính khả thi và hiệu quả của hướng tiếp cận dựa trên PSO đồng thời cũng mở ra tiềm năng ứng dụng của thuật toán ấy vào các bài toán định vị trong thực tế.

### LỜI CẢM ƠN

Xin chân thành cảm ơn Đại học Cần Thơ đã tài trợ và tạo điều kiện cho chúng tôi nghiên cứu đề tài này, mã số: TSV2025-184.

### TÀI LIỆU THAM KHẢO

Baskar, S., & Suganthan, P. N. (2004). A novel concurrent particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)* (pp. 792–796). IEEE. <https://doi.org/10.1109/CEC.2004.1330922>

Drezner, Z., & Wesolowsky, G. O. (1978). Facility location on a sphere. *Journal of the Operational Research Society*, 29(10), 997–1004. <https://doi.org/10.1057/jors.1978.213>

Güner, A. R., & Şevkli, M. (2008). A discrete particle swarm optimization algorithm for uncapacitated facility location problem. *Journal of Artificial Evolution and Applications*, 2008, Article 861512. <https://doi.org/10.1155/2008/861512>

Katz, I. N., & Cooper, L. (1980). Optimal location on a sphere. *Computers & Mathematics with Applications*, 6(2), 175–196. [https://doi.org/10.1016/0898-1221\(80\)90027-9](https://doi.org/10.1016/0898-1221(80)90027-9)

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN)* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>

Krohling, R. A. (2004). Gaussian swarm: A novel particle swarm optimization algorithm. In *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems* (Vol. 1, pp. 372–376). IEEE. <https://doi.org/10.1109/ICCIS.2004.1460443>

Krohling, R. A., & Coelho, L. D. S. (2006). PSO-E: Particle swarm with exponential distribution. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1428–1433). IEEE. <https://doi.org/10.1109/CEC.2006.1688445>

Kuhn, H. W., & Kuenne, R. E. (1962). An efficient algorithm for the numerical solution of the

- generalized Weber problem in spatial economics. *Journal of Regional Science*, 4(2), 21–33.  
<https://doi.org/10.1111/j.1467-9787.1962.tb00902.x>
- Marianov, V., & Eiselt, H. A. (2024). Fifty years of location theory – A selective review. *European Journal of Operational Research*, 318(3), 701–718. <https://doi.org/10.1016/j.ejor.2024.01.024>
- Nguyen, V. L., Kwok, N. M., & Ha, Q. P. (2024). Fermat–Weber location particle swarm optimization for cooperative path planning of unmanned aerial vehicles. *Applied Soft Computing*, 167, 112269.  
<https://doi.org/10.1016/j.asoc.2024.112269>
- Rubio-López, F., Rubio, O., & Vidaurre, R. U. (2023). The inverse Weber problem on the plane and the sphere. *Mathematics*, 11(24), 5000.  
<https://doi.org/10.3390/math11245000>
- Sevкли, M., Mamedsaidov, R., & Camci, F. (2014). A novel discrete particle swarm optimization for p-median problem. *Journal of King Saud University - Engineering Sciences*, 26(1), 11–19.  
<https://doi.org/10.1016/j.jksues.2012.09.002>
- Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 69–73). IEEE.  
<https://doi.org/10.1109/ICEC.1998.699146>
- Shih, H. (2015). Facility location decisions based on driving distances on spherical surface. *American Journal of Operations Research*, 5(5), 450–492.  
<https://doi.org/10.4236/ajor.2015.55037>
- Suzuki, A. (2019). *Big Triangle–Small Triangle Method for the Weber Problem on the Sphere*. In H. A. Eiselt & V. Marianov (Eds.), *Contributions to Location Analysis* (pp. 109–123). Springer.  
[https://doi.org/10.1007/978-3-030-19111-5\\_4](https://doi.org/10.1007/978-3-030-19111-5_4)
- Weber, A. (1909). *Über den Standort der Industrien. Erster Teil: Reine Theorie des Standorts*. Mohr (J.C.B. Mohr Siebeck).
- Weiszfeld, E. (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43, 355–386.
- Zheng, L., Xu, G., & Chen, W. (2024). Using improved particle swarm optimization algorithm for location problem of drone logistics hub. *Computers, Materials & Continua*, 78(1), 935–957.  
<https://doi.org/10.32604/cmc.2023.046006>