

BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT TRONG DẠY HỌC LÝ THUYẾT ĐỒ THỊ Ở TRƯỜNG PHỔ THÔNG: MỘT NGHIÊN CỨU TRI THỨC LUẬN LỊCH SỬ

Nguyễn Thị Nga,
Huỳnh Hữu Phước,
Trịnh Kim Mai,
Nguyễn Bảo Khang,
Phạm Thành Đạt⁺

Trường Đại học Sư phạm Thành phố Hồ Chí Minh
+Tác giả liên hệ • Email: datpt@hcmue.edu.vn

Article history

Received: 19/11/2024

Accepted: 23/4/2025

Published: 05/7/2025

Keywords

Epistemology, Historical analysis, Graph theory, Dijkstra's algorithm

ABSTRACT

This paper explores the historical development and educational implications of graph theory, emphasizing its origin from Euler's solution to the "Seven Bridges of Königsberg" problem in 1736. It categorizes shortest-path algorithms based on problem types, input graph characteristics, and basic techniques, highlighting classical algorithms and Dijkstra's improvements. The paper also compares the classical Dijkstra's approach using set operations with its modern adaptation utilizing labeling techniques, emphasizing their respective benefits and challenges in teaching. Through proposed teaching scenarios, the study aims to enhance students' conceptual understanding of shortest-path algorithms, emphasizing both computational efficiency and pedagogical clarity. The findings suggest modifications to teaching methods for improved student comprehension and algorithmic implementation.

1. Mở đầu

Sự phát triển của lý thuyết đồ thị (LTĐT) có thể được bắt đầu từ thế kỉ XX, khi đồ thị tách khỏi mô hình để trở thành một đối tượng toán học độc lập. Tuy nhiên, sự kiện được cho là đánh dấu sự ra đời của LTĐT lại đến từ việc Euler giải quyết bài toán "bảy cây cầu ở Königsberg" vào năm 1736. Sự kiện này, không chỉ đánh dấu sự ra đời của LTĐT mà còn mở ra một lĩnh vực nghiên cứu quan trọng trong LTĐT: Tìm đường đi trên đồ thị. Ban đầu, mối quan tâm của các nhà toán học đối với vấn đề này chủ yếu xuất phát từ việc giải các câu đố toán học (Biggs et al., 1986). Với sự phát triển của xã hội, việc tìm đường đi trong đồ thị đã trở thành một bài toán tối ưu, cùng nhiều ứng dụng thực tiễn hơn, điển hình là bài toán tìm đường đi ngắn nhất trong đồ thị.

Thiết kế của nghiên cứu được thực hiện theo phương pháp phân tích tri thức luận lịch sử, đây là phương pháp tiếp cận với một đối tượng tri thức từ lịch sử hình thành và phát triển cho chính đối tượng đó. "Phân tích tri thức luận lịch sử một tri thức là một phân tích quá khứ để khám phá những mò mẫm, những lệch lạc, những chướng ngại khác nhau, những điều kiện cho phép xuất hiện các khái niệm khoa học mới. Trong phân tích tri thức luận lịch sử, điều kiện cho sự nảy sinh một phát minh cũng quan trọng không kém bản thân phát minh đó. Phân tích này giúp ta hiểu đầy đủ hơn sự tiến triển của tri thức, từ đó hiểu rõ hơn các hiện tượng dạy học tri thức đang bàn đến" (Lê Thị Hoài Châu, 2017, tr 20).

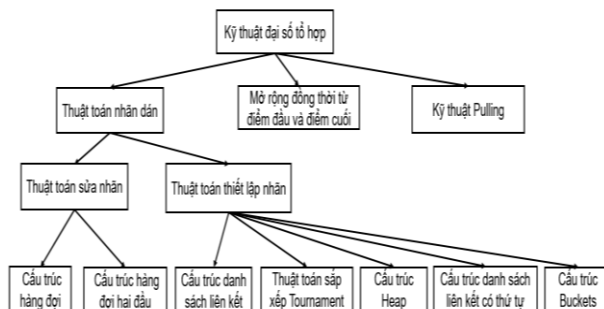
Một nghiên cứu tri thức luận về "Bài toán đường đi ngắn nhất" trong dạy học LTĐT ở phổ thông sẽ cho thấy tầm quan trọng của việc tiếp cận sự phát triển của thuật toán tìm đường đi ngắn nhất (đặc biệt là thuật toán Dijkstra) dưới góc nhìn của thế giới Toán học trong lịch sử. Thay vì chỉ xem xét các thuật toán tối ưu như những công cụ toán học thuần túy, nghiên cứu này làm sáng tỏ nguồn gốc, động cơ và quá trình hình thành các khái niệm cùng phương pháp giải quyết vấn đề. Qua đó, GV có thêm những tham chiếu khoa học khi thiết kế một tình huống dạy học giúp HS của mình không chỉ hiểu thuật toán Dijkstra và vận hành nó như một quy trình tính toán, mà còn thấy được con đường tư duy, những khó khăn, chướng ngại, điều kiện và bối cảnh lịch sử dẫn đến sự ra đời của các phiên bản, từ cổ điển đến cải tiến gần đây.

2. Kết quả nghiên cứu

2.1. Một số lý thuyết

2.1.1. Phân loại các lớp thuật toán tìm đường đi ngắn nhất

Theo Deo và Pang (1984), có thể phân loại các lớp thuật toán tìm đường ngắn nhất theo 3 cách: (1) Phân loại theo loại câu hỏi được đặt ra với một đồ thị cho trước; (2) Phân loại theo các đặc trưng nổi bật của đồ thị đầu vào; (3) Phân loại dựa trên các kỹ thuật nền tảng được sử dụng để giải quyết vấn đề tìm đường đi ngắn nhất. Cách phân loại thứ ba dựa trên các kỹ thuật nền tảng được sử dụng để giải quyết vấn đề, đóng vai trò quan trọng trong việc lựa chọn phương pháp tiếp cận hiệu quả nhất. Nổi bật trong cách phân loại này là lớp các thuật toán sử dụng phương pháp gán nhãn (Labeling Algorithm), được xếp vào nhóm các kỹ thuật tổ hợp - Combinatorial Technique (hình 1). Các thuật toán gán nhãn có thể được chia thành hai loại chính: (1) Thuật toán sửa nhãn (Label-Correcting algorithms); (2) Thuật toán thiết lập nhãn (Label-Setting algorithms). Hai ví dụ tiêu biểu đại diện cho hai loại thuật toán này là thuật toán Bellman-Ford và thuật toán Dijkstra.



Hình 1. Sơ đồ phân loại các kỹ thuật đại số để giải quyết vấn đề tìm đường đi ngắn nhất (Deo & Pang, 1984)

2.1.2. Thuật toán Dijkstra cổ điển

Vào năm 1956, Dijkstra ra mắt máy tính ARMAC với chương trình tìm đường đi ngắn nhất giữa 64 thành phố ở Hà Lan. Chương trình này vận hành dựa trên thuật toán do chính ông phát minh, được công bố vào năm 1959, thuật toán Dijkstra sau này trở thành nền tảng cho nhiều ứng dụng hiện đại như hệ thống định tuyến và bản đồ GPS, đánh dấu quan trọng của ông trong lịch sử khoa học máy tính (Frana & Misa, 2010). Dijkstra đã giới thiệu hai vấn đề liên quan đến việc tìm một đường đi ngắn nhất trong đồ thị đó là: (1) Xây dựng một cây có tổng độ dài nhỏ nhất giữa n đỉnh (một cây là một đồ thị với một và chỉ một đường đi giữa mọi cặp đỉnh); (2) Tìm đường đi có tổng độ dài nhỏ nhất giữa hai đỉnh cho trước P và Q. Trong thực tế, vấn đề 2 có tính ứng dụng cao hơn, đặc biệt trong các lĩnh vực như giao thông, viễn thông hay mạng xã hội. Vì vậy, nội dung tiếp theo sẽ tập trung phân tích thuật toán giải quyết vấn đề 2 - thuật toán tìm đường đi ngắn nhất giữa hai đỉnh, tiêu biểu là thuật toán Dijkstra. Ý tưởng cốt lõi trong thuật toán Dijkstra được phát biểu ngắn gọn như sau: Nếu một đỉnh R nằm trên đường đi ngắn nhất từ đỉnh P đến đỉnh Q, thì nhất thiết phải biết được đường đi ngắn nhất từ P đến R. Do đó, thuật toán của ông xây dựng đường đi ngắn nhất từ P đến Q bằng cách lần lượt tìm đường đi ngắn nhất từ P đến các đỉnh khác, theo thứ tự tăng dần khoảng cách, cho tới khi đến đỉnh Q. Các bước lập chính khi triển khai thuật toán Dijkstra sẽ được trình bày cụ thể ở như sau:

Dijkstra chia các đỉnh thành ba tập hợp A, B, C và các cạnh thành ba tập hợp I, II, III.

Các tập hợp đỉnh: Tập A - Các đỉnh nằm trong đường đi tối thiểu (các đỉnh này sẽ được thêm vào theo thứ tự tăng dần khoảng cách đến đỉnh); Tập B - Các đỉnh mà từ đó có thể được lựa chọn để đem sang tập A (tập này bao gồm tất cả các đỉnh có kết nối với ít nhất một đỉnh thuộc tập A); Tập C - Các đỉnh còn lại.

Các tập hợp cạnh: Tập I - Các cạnh xuất hiện trong đường đi ngắn nhất từ đỉnh P đến các đỉnh khác trong tập A; Tập II - Các cạnh mà từ đó có thể được lựa chọn để đem sang tập I (mỗi cạnh trong tập này chỉ dẫn đến một đỉnh trong tập B); Tập III - Các cạnh còn lại (bị bỏ hoặc chưa xem xét).

Sau khi đã phân loại các đỉnh và các cạnh thì ta sẽ bắt đầu đi vào các bước lập trong thuật toán. Ban đầu tất cả các đỉnh của đồ thị đều thuộc tập C và tất cả cạnh nằm trong tập III. Ta bắt đầu bằng cách chuyển đỉnh P sang tập A. Và lặp lại hai bước sau đây:

Bước 1. Xem xét tất cả các cạnh r kết nối đỉnh vừa được chuyển vào tập A với các đỉnh R trong tập B hoặc C.

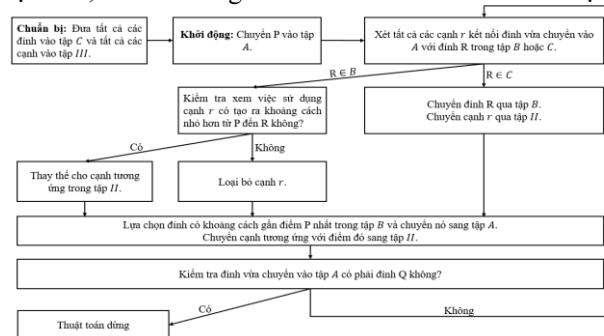
+ Nếu đỉnh R thuộc B thì ta xem xét xem việc sử dụng cạnh r có tạo ra đường đi ngắn hơn từ P đến R so với việc sử dụng các cạnh đã biết trong tập II hay không. Nếu không thì cạnh r bị loại bỏ, còn nếu việc sử dụng cạnh r tạo ra đường đi ngắn hơn thì ta sẽ thay thế bằng cạnh tương ứng trong tập II.

+ Nếu đỉnh R thuộc tập C thì ta sẽ thêm nó vào tập B và cạnh r sẽ được thêm vào tập II.

Bước 2. Mỗi đỉnh trong tập B chỉ có thể được kết nối đỉnh P bởi một cách (bởi các cạnh từ tập I và một cạnh từ tập II). Do đó mỗi đỉnh trong tập B đều có một khoảng cách duy nhất đối với P, lựa chọn đỉnh có khoảng cách nhỏ

nhất chuyển từ tập B sang tập A và cạnh tương ứng sẽ được chuyển từ tập II sang tập I . Khi này có một điểm mới được chuyển sang tập A cho nên ta quay lại bước 1 và lặp lại như vậy cho đến khi đỉnh Q được chuyển sang tập A .

Mô hình hoạt động của thuật toán Dijkstra cổ điển được minh họa trong sơ đồ (hình 2). Thuật toán này dựa trên nguyên tắc lần lượt lựa chọn các đỉnh theo thứ tự khoảng cách tăng dần từ đỉnh nguồn P . Điểm nổi bật của thuật toán Dijkstra cổ điển nằm ở cách tiếp cận thông minh, khi nó giúp rút ngắn thời gian tính toán bằng cách bỏ qua các đỉnh không đóng vai trò quan trọng trong việc tìm kiếm đường đi ngắn nhất. Ngoài ra cách mô tả về các bước lặp của thuật toán cũng giúp người đọc dễ dàng hình dung và nắm được ý tưởng của thuật toán. Tuy nhiên, việc triển khai thuật toán phụ thuộc nhiều vào biểu diễn tập hợp đã vô tình gây khó khăn cho người học khi khối lượng công việc cần làm sẽ rất nhiều và dễ gây nhầm lẫn khi phải liên tục di chuyển các đỉnh, cạnh qua lại giữa các tập hợp. Vì vậy, để cải thiện tính khả thi và hiệu suất, cần có những điều chỉnh và tối ưu hóa cho thuật toán Dijkstra cổ điển.



Hình 2. Sơ đồ minh họa triển khai thuật toán Dijkstra cổ điển

2.1.3. Thuật toán Dijkstra cải tiến

Thuật toán Dijkstra mà chúng ta được học ngày nay đã được phát triển và cải tiến dựa trên phương pháp gán nhãn (Labeling Methods), tạo nên sự khác biệt đáng kể so với cách tiếp cận ban đầu. Ngoại trừ thuật toán Dijkstra cổ điển, được coi là một khuôn mẫu cơ bản, không có thuật toán Dijkstra cải tiến nào được xem là chính thức. Nguyên nhân là vì tất cả các thuật toán cải tiến này đều dựa trên một kỹ thuật chung: gán nhãn. Do đó, có nhiều cách mô tả khác nhau về việc triển khai thuật toán. Chẳng hạn, một số triển khai có thể lưu giữ các nhãn cho các đỉnh trên hình vẽ đồ thị (Cormen et al., 2022), trong khi những triển khai khác lại lưu giá trị các nhãn trong bảng (Javaid, 2013). Dựa vào các tài liệu đã đề cập, chúng tôi trình bày một mô tả tổng quát về các phiên bản cải tiến của thuật toán Dijkstra, mà chúng tôi gọi là thuật toán Dijkstra cải tiến.

Thuật toán Dijkstra cải tiến: Mở đầu: Gán nhãn cho đỉnh A bằng 0 và cho tất cả các đỉnh khác bằng ∞ . Khoanh tròn đỉnh A . Các bước lặp, trong mỗi bước lặp, thực hiện các thao tác sau đây: Gọi U là đỉnh vừa khoanh tròn ở bước trước, trong số các đỉnh chưa khoanh tròn, xét lần lượt từng đỉnh V kề với đỉnh U , tính giá trị $n_U + w_{UV}$, sau đó so sánh giá trị này với nhãn hiện tại n_V của đỉnh V (Nếu giá trị đó nhỏ hơn, cập nhật nhãn n_V thành giá trị mới); So sánh nhãn của tất cả các đỉnh chưa khoanh tròn. Khoanh tròn đỉnh nào có nhãn nhỏ nhất (nếu có nhiều đỉnh có nhãn nhỏ nhất, khoanh tròn một đỉnh tùy ý trong số đó); Nếu đỉnh T chưa được khoanh tròn, thực hiện bước lặp tiếp theo, nếu không, kết thúc các bước lặp.

Kết luận: Dò lại các bước lặp để ghi lại nhãn n_T của đỉnh T dưới dạng tổng độ dài các cạnh. Từ đó, xác định được đường đi ngắn nhất từ A đến T cùng với độ dài của nó.

Trong bài báo này, chúng tôi sử dụng thuật ngữ “gán nhãn” để chỉ những cải tiến dựa trên thuật toán Dijkstra cổ điển. Thuật toán Dijkstra ban đầu không đề cập đến phương pháp gán nhãn, nhưng tư tưởng này dần xuất hiện trong các biến thể nhằm khắc phục hạn chế khi lập trình trên máy tính. Phương pháp gán nhãn ra đời như một giải pháp hình thức giúp tối ưu hóa việc lập trình, giảm khối lượng xử lý và làm cho thuật toán dễ triển khai hơn. Ford (1956) là người được ghi nhận đầu tiên áp dụng gán nhãn trong giải bài toán đường đi ngắn nhất, đặt nền móng cho thuật toán Bellman-Ford sau này.

Công trình của Ford (1956): Cho một mạng (đồ thị), kí hiệu các đỉnh là P_i với $i=0,1,\dots,N$ (P_0 là điểm gốc, P_N là điểm cuối) và các cạnh nối từ P_i đến P_j được kí hiệu là A_{ij} . Liên kết cạnh A_{ij} với một dung lượng c_{ij} và một độ dài (hoặc chi phí) l_{ij} . Sau đó ông đặt ra ba vấn đề: (1) Tìm lưu lượng tối đa từ P_0 đến P_N (độc lập với chi phí); (2) Tìm đường đi ngắn nhất từ P_0 đến P_N (độc lập với lưu lượng); (3) Tìm lượng hàng hoá tối đa vận chuyển từ P_0 đến P_N trong một khoảng thời gian cho trước (ở vấn đề này ta hiểu l_{ij} là thời gian cần thiết để vận chuyển hàng hoá từ P_0 đến P_N). Ở đây chúng ta chỉ quan tâm tới vấn đề (2), tức là vấn đề tìm đường đi ngắn nhất

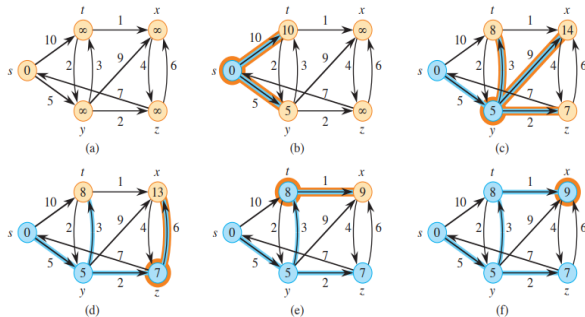
từ P_0 đến P_N . Đối với loại vấn đề này, Ford gọi x_i là khoảng cách ngắn nhất từ P_i đến P_N và đưa ra giải pháp như sau:

Gán $x_0=0$ và $x_i=\infty$ với $i \neq 0$. Quét trong mạng để tìm các cặp P_i và P_j có tính chất $x_i - x_j > l_{ij}$. Đối với cặp này ta sẽ thay thế x_j bởi $x_i + l_{ij}$. Tiếp tục quá trình đó đến khi không thể tìm thấy các cặp như vậy nữa và x_N lúc này nhỏ nhất đồng thời mô tả khoảng cách ngắn nhất từ P_0 đến P_N .

Công trình của Ford (1956) còn một số hạn chế khi chưa đưa ra được các hướng dẫn cụ thể trong việc lựa chọn các cặp P_i và P_j . Tuy nhiên, công trình này đã góp phần chuẩn hoá các thuật toán gán nhãn tìm đường đi ngắn nhất, nổi bật nhất trong đó có lẽ là thuật toán Dijkstra cải tiến.

2.2. So sánh thuật toán Dijkstra cổ điển và thuật toán Dijkstra cải tiến

Mặc dù, hai thuật toán Dijkstra cổ điển và thuật toán Dijkstra cải tiến có nhiều điểm khác nhau về cấu trúc nhưng đều mang cùng một tư tưởng ban đầu của Dijkstra đó là: đường đi ngắn nhất từ P đến Q sẽ được xây dựng bằng cách tìm các đường đi ngắn nhất từ P đến các đỉnh khác theo thứ tự tăng dần khoảng cách cho đến khi đạt được đỉnh Q . Về mặt hình thức, hai thuật toán này hoàn toàn khác nhau. Đối với thuật toán Dijkstra cổ điển thì Dijkstra đã sử dụng các tập hợp để phân loại các tập hợp đỉnh và tập hợp cạnh, sau mỗi bước lặp ông sẽ chuyển các đỉnh và cạnh sang các tập hợp tương ứng cho đến khi đỉnh Q được chuyển sang tập A . Như vậy, thuật toán Dijkstra cổ điển giúp người học dễ dàng nhận ra ý tưởng bản chất của thuật toán. Tuy nhiên, thuật toán này lại có những hạn chế khi được triển khai dựa trên việc sử dụng các tập hợp. Cụ thể, sau mỗi bước lặp, các đỉnh và cạnh của đồ thị được chuyển vào các tập hợp tương ứng, cho đến khi đỉnh Q được chuyển vào tập hợp A . Điều này cho thấy rằng thông tin được lưu trữ sau mỗi bước lặp chính là các đỉnh và cạnh của đồ thị. Như vậy, việc triển khai thuật toán Dijkstra cổ điển có mối liên hệ mật thiết với việc sử dụng hình vẽ đồ thị. Ngoài việc dễ gây nhầm lẫn cho người học thì cách tiếp cận này còn gặp nhiều khó khăn khi số lượng đỉnh và cạnh của đồ thị được tăng lên. Thuật toán Dijkstra cải tiến được xây dựng dựa trên phương pháp gán nhãn, ở mỗi bước lặp thì người học chỉ cần lưu lại các nhãn trực tiếp trên hình vẽ đồ thị (hình 3) hoặc lưu lại nhãn dưới dạng bảng (hình 4) mà không cần phải thông qua bất cứ biểu diễn tập hợp nào.



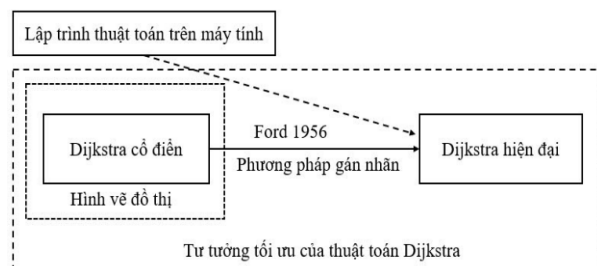
Hình 3. Triển khai thuật toán Dijkstra cải tiến dựa vào hình vẽ đồ thị (Cormen et al., 2022)

Visited	Node						
	B	C	D	E	F	G	
A	2	4	5	-	-	-	
A B	2	4	4	6	-	9	
A B C	2	4	4	6	7	9	
A B C D	2	4	4	5	7	9	
A B C D E		2	4	4	5	7	7
A B C D E F			2	4	4	5	7
A B C D E F G			2	4	4	5	7
Next	B	C	B	B	C	B	

Hình 4. Triển khai thuật toán Dijkstra dựa vào bảng (Javaid, 2013)

Như vậy, việc triển khai thuật toán Dijkstra cải tiến hoàn toàn có thể tách rời khỏi hình vẽ. Chính cách tiếp cận này đã tạo nên điểm mạnh của thuật toán Dijkstra cải tiến nói riêng và lớp các thuật toán gán nhãn nói chung, đó là sự gọn gàng và tiện lợi trong việc lưu trữ và cập nhật khoảng cách. Tuy nhiên, sự tiện lợi này vô tình làm ẩn đi ý tưởng của thuật toán và gây ra tình trạng người học có thể triển khai thuật toán mà không hiểu rõ ý nghĩa của từng bước lặp. Dù hiện nay khó có thể tìm thấy tài liệu nào đề cập đến thuật toán Dijkstra cổ điển, tuy nhiên mối liên hệ chặt chẽ và hỗ trợ lẫn nhau giữa hai phiên bản này đã được lịch sử ghi nhận và chúng ta có thể mô tả lại mối quan hệ này dưới dạng sơ đồ (hình 5).

2.3. Giới thiệu một tình huống dạy học bài toán tìm đường đi ngắn nhất



Hình 5. Sơ đồ thể hiện mối quan hệ giữa thuật toán Dijkstra cổ điển và thuật toán Dijkstra cải tiến

Mục tiêu xây dựng tình huống: Tình huống dạy học được xây dựng nhằm đáp ứng hai yêu cầu trong Chương trình giáo dục phổ thông 2018 (Bộ GD-ĐT, 2018) ở chuyên đề “Làm quen với một vài yếu tố của LTĐT”, gồm: (1) Nhận biết thuật toán tìm đường đi tối ưu trong các trường hợp đơn giản; (2) Vận dụng kiến thức đồ thị để giải quyết bài toán thực tiễn. Dựa trên phân tích sự chuyển đổi từ thuật toán Dijkstra cổ điển sang cải tiến, nhóm nghiên cứu thiết kế tình huống thực nghiệm giúp HS trải nghiệm quá trình hình thành và phát triển của thuật toán này. Thực nghiệm triển khai với HS lớp 10 và 11 (10 HS, chia thành 4 nhóm) trong học kỳ 2 năm học 2023-2024. Ở hoạt động đầu, HS mô phỏng đồ thị với đỉnh là quốc gia, cạnh là đường bay và trọng số là chi phí di chuyển. Các em liệt kê tất cả các đường đi từ Việt Nam đến Hà Lan, qua đó nhận thấy ưu điểm của việc biểu diễn bằng đồ thị. Sang hoạt động thứ hai, số lượng quốc gia tăng lên, khiến HS gặp khó khăn khi liệt kê thủ công toàn bộ đường đi. Đây là cơ hội để HS nhận ra sự tối ưu trong thuật toán Dijkstra cải tiến. Thông qua thiết kế câu hỏi và gợi mở, HS dần hình thành được các bước cơ bản của thuật toán, từ lưu trữ thông tin, sắp xếp đỉnh theo khoảng cách ngắn nhất, đến tách rời khỏi biểu diễn đồ thị ban đầu. Kết quả cho thấy tình huống này vừa tái hiện được quá trình phát triển của thuật toán, vừa hỗ trợ HS hiểu sâu bản chất và hình thành tư duy thuật toán một cách tự nhiên.

- **Hoạt động 1.** HS quan sát bảng được cho và phân tích yêu cầu trong phiếu học tập 1:

Bài toán 1. Hà Lan là đất nước mà An đã mong muốn được đặt chân đến từ lâu. Do dịch bệnh COVID-19 đã trở lại nên những chuyến bay của các hãng hàng không cũng bị hạn chế đi rất nhiều. Cụ thể, các hãng hàng không quốc tế có các chuyến bay (cùng với giá tiền, đơn vị: triệu đồng) như sau:

	Việt Nam	Hà Lan	Pháp	Bỉ	Thụy Điển
Việt Nam	X	X	18	30	23
Hà Lan	X	X	X	X	X
Pháp	X	38	X	9	12
Bỉ	X	11	9	X	9
Thụy Điển	X	23	18	3	X

Các nước khởi hành ở bảng màu lam (cột dọc) và các nước đích đến ở bảng màu lục (hàng ngang).

Sau khi đã thu thập được thông tin ở bảng trên, An nhận thấy là không phải luôn có chuyến bay trực tiếp từ Việt Nam đến Hà Lan, do đó bạn ấy sẽ phải mua vé từ Việt Nam đến các nước trung gian, sau đó từ một nước trung gian đến Hà Lan. Em hãy giúp An tìm ra chặng bay tiết kiệm chi phí nhất.

Mục tiêu hoạt động 1: HS có thể vẽ được đồ thị mô tả bảng được cho với đỉnh là các quốc gia, cạnh là đường nối giữa hai quốc gia nếu có chuyến bay trực tiếp và trọng số (giá tiền của chuyến bay) được gán trên mỗi cạnh (nếu có); HS có thể chỉ ra được nhiều hơn một đường đi khác nhau từ nơi xuất phát là Việt Nam và điểm đến là Hà Lan; HS có thể tính được độ dài của các đường đi (chi phí) từ Việt Nam đến Hà Lan; HS có thể so sánh và đưa ra được đường đi có độ dài (chi phí) nhỏ nhất từ những đường đi đã tìm.

- **Hoạt động 2.** GV bổ sung thêm thông tin vào bảng nhằm mục đích tăng độ phức tạp cho tình huống. Trong hoạt động này, HS cần phải đưa ra được một đường đi từ Hà Lan về Việt Nam với chi phí tương ứng. Để HS hình thành bước lập trong thuật toán Dijkstra và sau đó áp dụng được thuật toán Dijkstra để hoàn thành bài toán, HS cần hoàn thành hai yêu cầu tiếp theo trong phiếu học tập số 2.

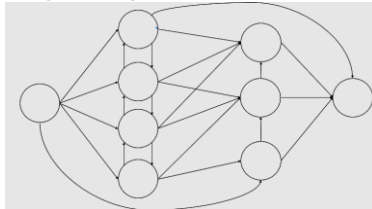
Bài toán 2. Sau khi đã kết thúc kì nghỉ hè tại Hà Lan, An lại tiếp tục lên kế hoạch cho các chuyến bay của mình để quay về Việt Nam. Do ở Hà Lan có nhiều chuyến bay hơn cho nên ta thu thập được thông tin các chuyến bay trong ngày hôm đó như sau: (Đơn vị: triệu đồng).

	Việt Nam	Hà Lan	Pháp	Bỉ	Thụy Điển	Nhật	Hàn	Mỹ	Anh
Việt Nam	X	X	X	X	X	X	X	X	X
Hà Lan	X	X	15	36	44	25	X	X	33
Pháp	X	X	X	37	24	X	28	33	X
Bỉ	X	X	2	X	X	9	5	20	X
Thụy Điển	23	X	11	X	X	X	4	X	X
Nhật	X	X	X	10	X	X	X	22	20
Hàn	21	X	X	X	X	X	X	X	X
Mỹ	15	X	X	X	X	X	19	X	X
Anh	35	X	X	X	X	X	X	15	X

Các nước khởi hành ở bảng màu lam (cột dọc) và các nước đích đến ở bảng màu lục (hàng ngang).

Em hãy giúp An thiết lập một lộ trình bay từ Hà Lan về Việt Nam sao cho tối ưu nhất về chi phí.

Yêu cầu 1: Hãy mô tả lại lộ trình và giá các chuyến bay vào sơ đồ bên dưới với các đỉnh là tên các quốc gia, các cạnh là các chuyến bay có thể giữa hai quốc gia, trọng số của các cạnh là chi phí khi bay giữa 2 quốc gia được kết nối.



Yêu cầu 2: Hãy liệt kê các địa điểm theo thứ tự: có chi phí tăng dần khi bay từ Hà Lan tới các địa điểm đó.

Lưu ý: Khi liệt kê các địa điểm trên hãy lưu lại địa điểm mà ta dừng chân trước đó.

Giả sử chi phí thấp nhất khi đi từ Hà Lan tới A là 100, đến B là 130, đến C là 140, đến D là 170,... Khi đó C là điểm có chi phí thấp nhất thứ 3 khi đi từ Hà Lan tới Việt Nam tuy nhiên lộ trình tối ưu nhất khi đi từ Hà Lan tới C không nhất thiết phải đi qua A và B mà có thể là đi trực tiếp từ Hà Lan đến C hoặc đi từ Hà Lan tới A rồi từ A đi thẳng đến C mà không cần thông qua B. Khi đó, ta lưu lại 2 thông tin trên C. Một là chi phí thấp nhất khi đi từ Hà Lan đến C là 140, thứ hai địa điểm ta đến trước đó trước khi tới C. Ta có thể ghi C như sau: C(140;A) trong đó A là địa điểm trước khi đi tới C trong lộ trình tối ưu từ Hà Lan tới C. Làm tương tự khi đến được Việt Nam.

Câu hỏi: + Việt Nam là nước gần Hà Lan thứ mấy?

+ Thông tin ta lưu tại Việt Nam vào thời điểm đó là gì?

+ Lộ trình tối ưu khi đi từ Hà Lan tới Việt Nam là gì?

Sau khi HS hoàn thành hai yêu cầu trên, GV sẽ đặt một số câu hỏi để khái quát hóa kiến thức cần dạy (thuật toán Dijkstra), chẳng hạn như: “Chúng ta có thể chắc chắn lộ trình trên là tối ưu không? Tại sao? (nghĩa là nếu như chúng ta liệt kê tất cả các lộ trình ra thì sẽ không có lộ trình nào tối ưu hơn đúng không?)”, “Theo em, mục tiêu của các bước lập ở yêu cầu 2 là gì?”, “Em hãy thử đưa ra các bước làm tổng quát để tìm đường đi tối ưu từ đỉnh này đến khác trong đồ thị có trọng số (dương)”.

Mục tiêu hoạt động 2: HS không thể đưa ra câu trả lời chính xác cho bài toán 2 khi vẫn sử dụng hình vẽ đồ thị; HS cần xây dựng thuật toán (theo hướng dẫn ở yêu cầu 2) để có thể đưa ra được câu trả lời đúng cho bài toán.

3. Kết luận

Từ việc phân tích lịch sử phát triển của thuật toán Dijkstra, nghiên cứu khẳng định tầm quan trọng của việc kết hợp cả phiên bản cổ điển và cải tiến trong giảng dạy bài toán tìm đường đi ngắn nhất. Trong đó, phiên bản cổ điển giúp HS nắm vững bản chất khái niệm, còn phiên bản cải tiến phát huy tính hiệu quả khi vận hành trên máy tính. Tuy nhiên, sự gia tăng mức độ trừu tượng trong các phiên bản cải tiến đòi hỏi cần có phương pháp sư phạm phù hợp để tránh làm mờ đi ý tưởng cốt lõi của thuật toán. Việc tích hợp phương pháp dạy học toán thông qua thực hành trải nghiệm và mô hình hoá toán học được đề xuất như một hướng tiếp cận hiệu quả, giúp HS vừa hiểu sâu lý thuyết, vừa rèn luyện khả năng phân tích, xây dựng và kiểm chứng mô hình trong bối cảnh thực tiễn. Đồng thời, việc phát triển các công cụ trực quan hóa, phần mềm mô phỏng và mở rộng nghiên cứu sang các bài toán phức tạp hơn được xác định là những định hướng quan trọng cho nghiên cứu và giảng dạy trong tương lai.

Tài liệu tham khảo

- Biggs, N., Lloyd, E. K., & Wilson, R. J. (1986). *Graph Theory, 1736-1936*. Oxford University Press.
- Bộ GD-ĐT (2018). *Chương trình giáo dục phổ thông môn Toán* (ban hành kèm theo Thông tư số 32/2018/TT-BGDĐT ngày 26/12/2018 của Bộ trưởng Bộ GD-ĐT).
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms*. MIT Press.
- Deo, N., & Pang, C. Y. (1984). Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2), 275-323.
- Ford, L. R. (1956). *Network flow theory*. Rand Corporation Paper, Santa Monica.
- Frana, P. L., & Misa, T. J. (2010). An interview with Edsger W. Dijkstra. *Communications of the ACM*, 53(8), 41-47.
- Javaid, A. (2013). *Understanding Dijkstra's algorithm*. Available at SSRN 2340905.
- Lê Thị Hoài Châu (2017). *Sự cần thiết của phân tích tri thức luận đối với các nghiên cứu về hoạt động dạy học và đào tạo giáo viên*. Kì yếu Hội thảo quốc tế về Didactic Toán lần thứ 6 (Actesdu sixieme colloque International en Didactique des Mathematiques). NXB Đại học Sư phạm Thành phố Hồ Chí Minh.