

MỘT SỐ BÀI TOÁN TÌM ĐƯỜNG TRÊN ĐỒ THỊ

5.1. ĐỒ THỊ CÓ TRỌNG SỐ VÀ BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT.

5.1.1. Mở đầu:

Trong đời sống, chúng ta thường gặp những tình huống như sau: đi từ địa điểm A đến địa điểm B trong thành phố, có nhiều đường đi, nhiều cách đi; có lúc ta chọn đường đi ngắn nhất (theo nghĩa cổ lý), có lúc lại chọn đường đi nhanh nhất (theo nghĩa thời gian) và có lúc phải cân nhắc để chọn đường đi rẻ tiền nhất (theo nghĩa chi phí), v.v...

Có thể coi sơ đồ của đường đi từ A đến B trong thành phố là một đồ thị, với đỉnh là các giao lộ (A và B coi như giao lộ), cạnh là đoạn đường nối hai giao lộ. Trên mỗi cạnh của đồ thị này, ta gán một số đường, có thể là chiều dài của đoạn đường, thời gian đi đoạn đường hoặc chi phí vận chuyển trên đoạn đường đó, ...

Đồ thị có trọng số là đồ thị $G=(V,E)$ mà mỗi cạnh (họ các cung) $e \in E$ được gán bởi một số thực $m(e)$, gọi là trọng số của cạnh (họ các cung) e .

Trong phần này, trọng số của mỗi cạnh được xét là một số đường và còn gọi là chiều dài của cạnh đó. Mỗi đường đi từ đỉnh u đến đỉnh v , có chiều dài là $m(u,v)$, bằng tổng chiều dài các cạnh mà nó đi qua. Khoảng cách $d(u,v)$ giữa hai đỉnh u và v là chiều dài đường đi ngắn nhất (theo nghĩa $m(u,v)$ nhỏ nhất) trong các đường đi từ u đến v .

Có thể xem một đồ thị G bất kỳ là một đồ thị có trọng số mà mỗi cạnh đều có chiều dài 1. Khi đó, khoảng cách $d(u,v)$ giữa hai đỉnh u và v là chiều dài của đường đi từ u đến v ngắn nhất, tức là đường đi qua ít cạnh nhất.

5.1.2. Bài toán tìm đường đi ngắn nhất:

Cho đồ thị liên thông, có trọng số $G=(V,E)$. Tìm khoảng cách $d(u_0,v)$ từ một đỉnh u_0 cho trước đến một đỉnh v bất kỳ của G và tìm đường đi ngắn nhất từ u_0 đến v .

Có một số thuật toán tìm đường đi ngắn nhất; ở đây, ta có thuật toán do E. Dijkstra, nhà toán học người Hà Lan, đề xuất năm 1959. Trong phiên bản mà ta sẽ trình bày, người ta giả sử đồ thị là vô hướng, các trọng số là đường. Cần thay đổi đôi chút là có thể giải được bài toán tìm đường đi ngắn nhất trong đồ thị có hướng.

Phương pháp của thuật toán Dijkstra là: xác định tuần tự đỉnh có khoảng cách đến u_0 từ nhỏ đến lớn.

Trước tiên, đỉnh có khoảng cách đến a nhỏ nhất chính là a , với $d(u_0,u_0)=0$. Trong các đỉnh $v \neq u_0$, tìm đỉnh có khoảng cách k_1 đến u_0 là nhỏ nhất. Định lý này phải là một trong các định lý về u_0 . Giả sử đó là u_1 . Ta có:

$$d(u_0,u_1) = k_1.$$

Trong các đỉnh $v \neq u_0$ và $v \neq u_1$, tìm đỉnh có khoảng cách k_2 đến u_0 là nhỏ nhất. Đỉnh này phải là một trong các đỉnh kề với u_0 hoặc với u_1 . Gọi số đó là u_2 . Ta có:

$$d(u_0, u_2) = k_2.$$

Tiếp tục như trên, cho đến bao giờ tìm được khoảng cách từ u_0 đến một đỉnh v của G . Nếu $V = \{u_0, u_1, \dots, u_n\}$ thì:

$$0 = d(u_0, u_0) < d(u_0, u_1) < d(u_0, u_2) < \dots < d(u_0, u_n).$$

5.1.3. Thuật toán Dijkstra:

procedure *Dijkstra* ($G=(V,E)$ là đồ thị liên thông, có trọng số với trọng số dương)
 $\{G$ có các đỉnh $a=u_0, u_1, \dots, u_n=z$ và trọng số $m(u_i, u_j)$, với $m(u_i, u_j) = \infty$ nếu (u_i, u_j) không là một cạnh trong $G\}$

for $i := 1$ **to** n

$L(u_i) := \infty$

$L(a) := 0$

$S := V \setminus \{a\}$

$u := a$

while $S \neq \emptyset$

begin

for tất cả các đỉnh v thuộc S

if $L(u) + m(u, v) < L(v)$ **then** $L(v) := L(u) + m(u, v)$

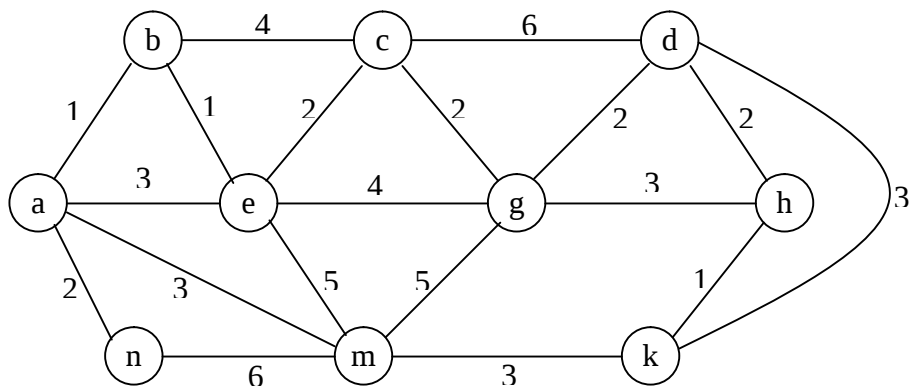
$u :=$ đỉnh thuộc S có nhãn $L(u)$ nhỏ nhất

$\{L(u):$ độ dài đường đi ngắn nhất từ a đến $u\}$

$S := S \setminus \{u\}$

end

Thí dụ 1: Tìm khoảng cách $d(a, v)$ từ a đến mỗi đỉnh v và tìm đường đi ngắn nhất từ a đến v cho trong đồ thị G sau.



L(a)	L(b)	L(c)	L(d)	L(e)	L(g)	L(h)	L(k)	L(m)	L(n)
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
—	1	∞	∞	3	∞	∞	∞	3	2
—	—	5	∞	2	∞	∞	∞	3	2
—	—	5	∞	2	∞	∞	∞	3	—
—	—	4	∞	—	6	∞	∞	3	—
—	—	4	∞	—	6	∞	6	—	—
—	—	—	10	—	6	∞	6	—	—
—	—	—	8	—	—	9	6	—	—
—	—	—	8	—	—	—	—	—	—
—	—	—	8	—	—	—	—	—	—

5.1.4. Định lý: Thuật toán Dijkstra tìm được đường đi ngắn nhất từ một đỉnh cho trước đến một đỉnh tùy ý trong đồ thị vô hướng liên thông có trọng số.

Chứng minh: Định lý được chứng minh bằng quy nạp. Tại bước k ta có giả thiết quy nạp là:

(i) Nhận vào đỉnh v không thuộc S là độ dài của đường đi ngắn nhất từ đỉnh a tới đỉnh này;

(ii) Nhận vào đỉnh v trong S là độ dài của đường đi ngắn nhất từ đỉnh a tới đỉnh này và đường đi này chỉ qua các đỉnh (ngoài chính đỉnh này) không thuộc S .

Khi $k=0$, tức là khi chưa có bước nào được thực hiện, $S=V \setminus \{a\}$, vì thế độ dài của đường đi ngắn nhất từ a tới các đỉnh khác a là ∞ và độ dài của đường đi ngắn nhất từ a tới chính nó bằng 0 (ở đây, chúng ta cho phép đường đi không có cạnh). Do đó bước cơ sở là đúng.

Giả sử giả thiết quy nạp là đúng với bước k . Gọi v là đỉnh lấy ra khỏi S ở bước $k+1$, vì vậy v là đỉnh thuộc S cuối cùng ở bước k có nhãn nhỏ nhất (nếu có nhiều đỉnh có nhãn nhỏ nhất thì có thể chọn một đỉnh nào đó làm v). Từ giả thiết quy nạp ta thấy rằng

trở lại khi vào vòng lặp thứ $k+1$, các đỉnh không thuộc S đã được gán nhãn bằng độ dài của đường đi ngắn nhất từ a . Đỉnh v cũng vậy phải được gán nhãn bằng độ dài của đường đi ngắn nhất từ a . Nếu điều này không xảy ra thì cấu trúc lặp thứ k sẽ có đường đi với độ dài nhỏ hơn $L_k(v)$ của các đỉnh thuộc S (vì $L_k(v)$ là độ dài của đường đi ngắn nhất từ a tới v chỉ các các đỉnh không thuộc S sau bước lặp thứ k). Giả sử là đỉnh đầu tiên của đường đi này thuộc S . Đó là đường đi với độ dài nhỏ hơn $L_k(v)$ từ a tới u chỉ các các đỉnh không thuộc S . Điều này trái với cách chọn v . Do đó (i) vẫn còn đúng cấu trúc lặp $k+1$.

Giả sử là đỉnh thuộc S sau bước $k+1$. Đường đi ngắn nhất từ a tới u chỉ các các đỉnh không thuộc S sẽ hoặc là chứa v hoặc là không. Nếu nó không chứa v thì theo giả thiết quy nạp độ dài của nó là $L_k(v)$. Nếu nó chứa v thì nó sẽ tạo thành đường đi từ a tới v với độ dài có thể ngắn nhất và chỉ các các đỉnh không thuộc S khác v , kết thúc bằng chính v tới u . Khi đó độ dài của nó sẽ là $L_k(v)+m(v,u)$. Điều đó chứng tỏ (ii) là đúng vì $L_{k+1}(u)=\min(L_k(u), L_k(v)+m(v,u))$.

5.1.5. Mệnh đề: Thuật toán Dijkstra tìm đường đi ngắn nhất từ một đỉnh cho trước đến một đỉnh tùy ý trong đồ thị vô hướng liên thông có trọng số có độ phức tạp là $O(n^2)$.

Chứng minh: Thuật toán dùng không quá $n-1$ bước lặp. Trong mỗi bước lặp, dùng không hơn $2(n-1)$ phép cộng và phép so sánh để sửa đổi nhãn của các đỉnh. Ngoài ra, một đỉnh thuộc S_k có nhãn nhỏ nhất nhất không quá $n-1$ phép so sánh. Do đó thuật toán có độ phức tạp $O(n^2)$.

5.1.6. Thuật toán Floyd:

Cho $G=(V,E)$ là đồ thị có hướng, có trọng số. Để tìm đường đi ngắn nhất giữa một cặp đỉnh của G , ta có thể áp dụng thuật toán Dijkstra nhiều lần hoặc áp dụng thuật toán Floyd để trình bày dưới đây.

Giả sử $V=\{v_1, v_2, \dots, v_n\}$ và có ma trận trọng số là $W \equiv W_0$. Thuật toán Floyd xây dựng dãy các ma trận vuông cấp n là W_k ($0 \leq k \leq n$) như sau:

procedure Xác định W_n

for $i := 1$ **to** n

for $j := 1$ **to** n

$W[i,j] := m(v_i, v_j)$ { $W[i,j]$ là phần tử dòng i cột j của ma trận W_0 }

for $k := 1$ **to** n

if $W[i,k] + W[k,j] < W[i,j]$ **then** $W[i,j] := W[i,k] + W[k,j]$

 { $W[i,j]$ là phần tử dòng i cột j của ma trận W_k }

5.1.7. Định lý: Thuật toán Floyd cho ta ma trận $W^*=W_n$ là ma trận khoảng cách nhỏ nhất của đồ thị G .

Chứng minh: Ta chứng minh bằng quy nạp theo k như sau:

$W_k[i, j]$ là chiều dài đường đi ngắn nhất trong nhúng đường đi nối đỉnh v_i và v_j đi qua các đỉnh trung gian trong $\{v_1, v_2, \dots, v_k\}$.

Trước hết minh chứng hiển nhiên đúng với $k=0$.

Giả sử mệnh đề đúng với $k-1$.

Xét $W_k[i, j]$. Có hai trường hợp:

1) Trong các đường đi chiều dài ngắn nhất nối v_i và v_j và đi qua các đỉnh trung gian trong $\{v_1, v_2, \dots, v_k\}$, có một đường đi γ sao cho $v_k \notin \gamma$. Khi đó γ cũng là đường đi ngắn nhất nối v_i và v_j đi qua các đỉnh trung gian trong $\{v_1, v_2, \dots, v_{k-1}\}$, nên theo giả thiết quy nạp,

$$W_{k-1}[i, j] = \text{chiều dài } \gamma \leq W_{k-1}[i, k] + W_{k-1}[k, j].$$

Do đó theo định nghĩa của W_k thì $W_k[i, j] = W_{k-1}[i, j]$.

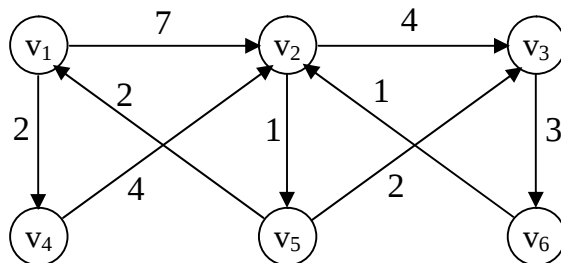
2) Mọi đường đi chiều dài ngắn nhất nối v_i và v_j và đi qua các đỉnh trung gian trong $\{v_1, v_2, \dots, v_k\}$, đều chứa v_k . Gọi $\gamma = v_i \dots v_k \dots v_j$ là một đường đi ngắn nhất nhất thì $v_i \dots v_k$ và $v_k \dots v_j$ cũng là những đường đi ngắn nhất đi qua các đỉnh trung gian trong $\{v_1, v_2, \dots, v_{k-1}\}$ và

$$\begin{aligned} W_{k-1}[i, k] + W_{k-1}[k, j] &= \text{chiều dài}(v_i \dots v_k) + \text{chiều dài}(v_k \dots v_j) \\ &= \text{chiều dài } \gamma < W_{k-1}[i, j]. \end{aligned}$$

Do đó theo định nghĩa của W_k thì ta có:

$$W_k[i, j] = W_{k-1}[i, k] + W_{k-1}[k, j].$$

Thí dụ 2: Xét đồ thị G sau:



Áp dụng thuật toán Floyd, ta tìm được (các ô trống là ∞)

$$W = W_0 = \begin{pmatrix} 7 & 2 & & & & \\ & 4 & 1 & & & \\ & & & 3 & & \\ 4 & & & & & \\ 2 & 2 & & & & \\ 1 & & & & & \end{pmatrix}$$

$$W_1 = \begin{pmatrix} 7 & 2 & & & & \\ & 4 & 1 & & & \\ & & & 3 & & \\ 4 & & & & & \\ 2 & 9 & 2 & 4 & & \\ 1 & & & & & \end{pmatrix}, W_2 = \begin{pmatrix} 7 & 11 & 2 & 8 & & \\ & 4 & 1 & & & \\ & & & 3 & & \\ 4 & 8 & & 5 & & \\ 2 & 9 & 2 & 4 & 10 & \\ 1 & 5 & 2 & & & \end{pmatrix}$$

$$W_3 = \begin{pmatrix} 7 & 11 & 2 & 8 & 14 & \\ & 4 & 1 & 7 & & \\ & & & 3 & & \\ 4 & 8 & & 5 & 11 & \\ 2 & 9 & 2 & 4 & 10 & 5 \\ 1 & 5 & 2 & 8 & & \end{pmatrix}, W_4 = \begin{pmatrix} 6 & 10 & 2 & 7 & 13 & \\ & 4 & 1 & 7 & & \\ & & & 3 & & \\ 4 & 8 & & 5 & 11 & \\ 2 & 8 & 2 & 4 & 9 & 5 \\ 1 & 5 & 2 & 8 & & \end{pmatrix}$$

$$W_5 = \begin{pmatrix} 9 & 6 & 9 & 2 & 7 & 12 & \\ 3 & 9 & 3 & 5 & 1 & 6 & \\ & & & & 3 & & \\ 7 & 4 & 7 & 9 & 5 & 10 & \\ 2 & 8 & 2 & 4 & 9 & 5 & \\ 4 & 1 & 4 & 6 & 2 & 7 & \end{pmatrix}, W^* = W_6 = \begin{pmatrix} 9 & 6 & 9 & 2 & 7 & 12 & \\ 3 & 7 & 3 & 5 & 1 & 6 & \\ 7 & 4 & 7 & 9 & 5 & 3 & \\ 7 & 4 & 7 & 9 & 5 & 10 & \\ 2 & 6 & 2 & 4 & 7 & 5 & \\ 4 & 1 & 4 & 6 & 2 & 7 & \end{pmatrix}.$$

Thuật toán Floyd có thể áp dụng cho đồ thị vô hướng cũng như đồ thị có hướng. Ta chỉ cần thay đổi cách vô hướng (u,v) bằng một cặp cạnh có hướng (u,v) và (v,u) với $m(u,v)=m(v,u)$. Tuy nhiên, trong trường hợp này, các phần tử trên đường chéo của ma trận W cần đặt bằng 0.

Đồ thị có hướng G là liên thông mạnh khi và chỉ khi mọi phần tử nằm trên đường chéo trong ma trận trọng số ngắn nhất W^* đều hữu hạn.

5.2. BÀI TOÁN LUỒNG CỰC ĐẠI.

5.2.1. Luồng cực đại:

5.2.1.1. Định nghĩa: Mạng vận tải là một đồ thị có hướng, không có khuyên và có trọng số $G=(V,E)$ với $V=\{v_0, v_1, \dots, v_n\}$ thỏa mãn:

- 1) Mọi cung $e \in E$ có trọng số $m(e)$ là một số nguyên không âm và đầu cuối là khả năng thông qua của cung e .
- 2) Có một và chỉ một đỉnh v_0 không có cung đi vào, tức là $\deg_i(v_0)=0$. Đỉnh v_0 được gọi là vào hay đỉnh phát của mạng.
- 3) Có một và chỉ một đỉnh v_n không có cung đi ra, tức là $\deg_o(v_n)=0$. Đỉnh v_n được gọi là ra hay đỉnh thu của mạng.

5.2.1.2. Định nghĩa: Một đường luồng khai thác, tức là xác định luồng vật chất chuyển qua mạng vận tải $G=(V,E)$, ngược lại ta đưa ra khái niệm luồng vận tải và nó được định nghĩa như sau.

Hàm φ xác định trên tập cung E và nhận giá trị nguyên được gọi là luồng vận tải của mạng vận tải G nếu φ thỏa mãn:

- 1) $\varphi(e) \geq 0, \forall e \in E$.
- 2) $\sum_{e \in \Gamma^-(v)} \varphi(e) = \sum_{e \in \Gamma^+(v)} \varphi(e), \forall v \in V, v \neq v_0, v \neq v_n$. Ở đây, $\Gamma^-(v) = \{e \in E \mid e \text{ có đỉnh cuối là } v\}$,
 $\Gamma^+(v) = \{e \in E \mid e \text{ có đỉnh đầu là } v\}$.
- 3) $\varphi(e) \leq m(e), \forall e \in E$.

Ta xem $\varphi(e)$ như là luồng hàng chuyển trên cung $e=(u,v)$ từ đỉnh u đến đỉnh v và không vượt quá khả năng thông qua của cung này. Ngoài ra, từ điều kiện 2) ta thấy rằng nếu $u \neq v$ không phải là vào v_0 hay ra v_n , thì luồng hàng chuyển tại v bằng luồng hàng chuyển khỏi v .

Từ quan hệ 2) suy ra:

$$4) \sum_{e \in \Gamma^+(v_0)} \varphi(e) = \sum_{e \in \Gamma^-(v_n)} \varphi(e) =: \varphi_{v_n}.$$

Đôi luồng φ_{v_n} (ta còn ký hiệu là φ_n) được gọi là luồng qua mạng, hay còn gọi là luồng tại điểm v_n hay giá trị của luồng φ . Bài toán đặt ra ở đây là tìm φ để φ_{v_n} đạt giá trị lớn nhất, tức là tìm giá trị lớn nhất của luồng.

5.2.1.3. Định nghĩa: Cho mạng vận tải $G=(V,E)$ và $A \subset V$. Ký hiệu

$$\Gamma^-(A) = \{(u,v) \in E \mid v \in A, u \notin A\}, \quad \Gamma^+(A) = \{(u,v) \in E \mid u \in A, v \notin A\}.$$

Đôi với tập cung M tùy ý, đôi luồng $\varphi(M) = \sum_{e \in M} \varphi(e)$ được gọi là luồng của tập cung M .

Từ điều kiện 2) dễ dàng suy ra hệ quả sau.

5.2.1.4. Hệ quả: Cho φ là luồng của mạng vận tải $G=(V,E)$ và $A \subset V \setminus \{v_0, v_n\}$. Khi đó:

$$\varphi(\Gamma^-(A)) = \varphi(\Gamma^+(A)).$$

5.2.2. Bài toán luồng cực đại:

Cho mạng vận tải $G=(V,E)$. Hãy tìm luồng φ để đạt φ_{v_n} max trên mạng G .

Nguyên lý của các thuật toán giải bài toán tìm luồng cực đại là như sau.

5.2.2.1. Định nghĩa: Cho $A \subset V$ là tập con tùy ý không chứa vào v_0 và chứa ra v_n . Tập $\Gamma^-(A)$ được gọi là mặt cắt đi của mạng vận tải G .

Đôi luồng $m(\Gamma^-(A)) = \sum_{e \in \Gamma^-(A)} m(e)$ được gọi là khả năng thông qua của mặt cắt đi $\Gamma^-(A)$.

Từ định nghĩa thiết diện và khả năng thông qua của nó ta nhận thấy rằng: mỗi đỉnh và hàng hoá được chuyển từ v_0 đến v_n ít nhất cũng phải đi qua một cung nào đó của thiết diện $\Gamma^-(A)$. Vì vậy, dù luồng φ và thiết diện $\Gamma^-(A)$ như thế nào đi nữa cũng vẫn tồn tại mối quan hệ:

$$\varphi_n \leq m(\Gamma^-(A)).$$

Do đó, nếu đi với luồng φ và thiết diện W mà có:

$$\varphi_n = m(W)$$

thì chắc chắn rằng luồng φ đạt giá trị lớn nhất và thiết diện W có khả năng thông qua nhỏ nhất.

5.2.2.2. Định nghĩa: Cung e trong mạng vận tải G với luồng vận tải φ được gọi là cung bão hoà nếu $\varphi(e) = m(e)$.

Luồng φ của mạng vận tải G được gọi là luồng đầy nếu mọi đường đi từ v_0 đến v_n đều chứa ít nhất một cung bão hoà.

Từ định nghĩa trên ta thấy rằng, nếu luồng φ trong mạng vận tải G chứa đầy thì nhất định tồn tại đường đi α từ v_0 vào và v_n không chứa cung bão hoà. Khi đó ta nâng luồng φ thành φ' như sau:

$$\varphi'(e) = \begin{cases} \varphi(e) + 1 & \text{khi } e \in \alpha, \\ \varphi(e) & \text{khi } e \notin \alpha. \end{cases}$$

Khi đó φ' cũng là một luồng, mà giá trị của nó là:

$$\varphi'_n = \varphi_n + 1 > \varphi_n.$$

Như vậy, đi với mọi luồng không đầy ta có thể nâng giá trị của nó và nâng cho tới khi nhận được một luồng đầy.

Tuy vậy, thực tế cho thấy rằng có thể có một luồng đầy, nhưng vẫn chưa đạt tới giá trị cực đại. Bởi vậy, cần phải dùng thuật toán Ford-Fulkerson để tìm giá trị cực đại của luồng.

5.2.2.3. Thuật toán Ford-Fulkerson:

Để tìm luồng cực đại của mạng vận tải G , ta xuất phát từ luồng tuỳ ý φ của G , rồi nâng luồng lên đầy, sau đó áp dụng thuật toán Ford-Fulkerson hoặc ta có thể áp dụng thuật toán Ford-Fulkerson trực tiếp đi với luồng φ .

Thuật toán gồm 3 bước:

Bước 1 (đánh dấu - định của mạng): Đi vào v_0 được đánh dấu bằng 0.

- 1) Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số $+i$ để đánh dấu cho mọi đỉnh y chứa được đánh dấu mà $(v_i, y) \in E$ và cung này chưa bão hoà ($\varphi(v_i, y) < m(v_i, y)$).
- 2) Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số $-i$ để đánh dấu cho mọi đỉnh z chứa được đánh dấu mà $(z, v_i) \in E$ và luồng của cung này dương ($\varphi(z, v_i) > 0$).

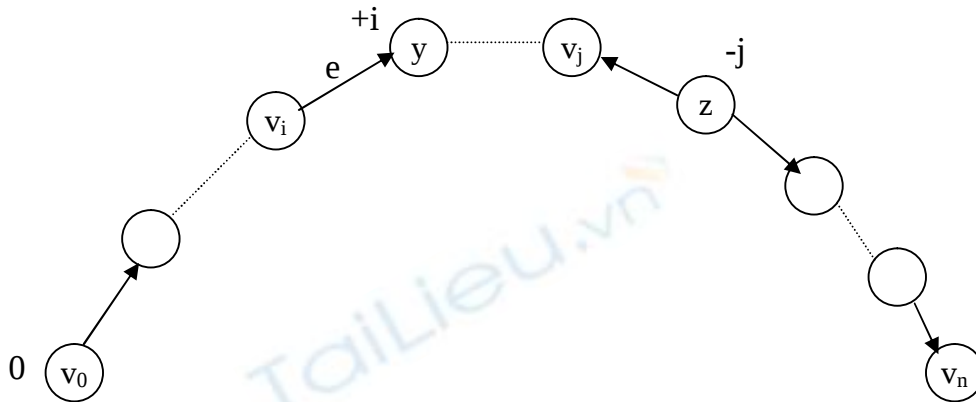
Nếu v là đỉnh đang xét thì ta đánh dấu đỉnh v là v_n thì trong G tồn tại g là v_0 và v_n mà t xích α , mà i đỉnh g khác nhau và g đánh dấu theo chiều α của đỉnh liên tiếp nó (chỉ sai khác nhau về dấu). Khi đó chỉ cần ta nâng giá trị của g .

Bước 2 (nâng giá trị của g): Để nâng giá trị của g là g , ta đặt:

$$g'(e) = g(e), \quad \text{nếu } e \notin \alpha,$$

$$g'(e) = g(e) + 1, \quad \text{nếu } e \in \alpha \text{ và } g \text{ đang xét theo chiều của xích } \alpha \text{ đi từ } v_0 \text{ đến } v_n,$$

$$g'(e) = g(e) - 1, \quad \text{nếu } e \in \alpha \text{ và } g \text{ đang xét ngược chiều của xích } \alpha \text{ đi từ } v_0 \text{ đến } v_n.$$



g' thỏa mãn các điều kiện về g , nên g' là một g và ta có:

$$g'_n = g_n + 1.$$

Như vậy, ta đã nâng giá trị của g lên một đơn vị.

Sau đó lặp lại một vòng mới. Vì khả năng thông qua của các cung g hữu hạn, nên quá trình phải dừng lại sau một số hữu hạn bước.

Bước 3: Nếu g là g^0 bằng phương pháp trên ta không thể nâng giá trị của g lên nữa, nghĩa là ta không thể đánh dấu đỉnh v_n , thì ta nói rằng quá trình nâng g kết thúc và g^0 đã đạt giá trị cực đại, đồng thời g^0 là g kết thúc.

Khi mạng v là $G=(V,E)$ đạt giá trị g^0 , thì bài toán tìm theo ta không thể đánh dấu đỉnh v là v_n . Trên cơ sở hiện trạng g đánh dấu bài toán này, ta sẽ chứng minh rằng g^0 đã đạt giá trị cực đại.

5.2.2.4. Bước 4: Cho g là mạng v là $G=(V,E)$ và $A \subset V$, chia v là v_n và không chia v là v_0 . Khi đó:

$$g_{v_n} = g(\Gamma^-(A)) - g(\Gamma^+(A)).$$

Chứng minh: Xét $A_1 = A \setminus \{v_n\}$. Theo Hq 5.2.1.4, ta có:

$$g(\Gamma^-(A_1)) = g(\Gamma^+(A_1)) \quad (1).$$

Xét $C_1 = \{(a, v_n) \in E \mid a \in A\}$. Khi đó $\Gamma^-(A) = \Gamma^-(A_1) \cup C_1$ và $\Gamma^-(A_1) \cap C_1 = \emptyset$, nên

$$g(\Gamma^-(A)) = g(\Gamma^-(A_1)) + g(C_1) \quad (2).$$

Xét $C_2 = \{(b, v_n) \in E \mid b \in A_1\}$. Khi đó $C_2 = \{(b, v_n) \in E \mid b \in A_1\}$, $\Gamma^+(A_1) = \Gamma^+(A) \cup C_2$ và $\Gamma^+(A) \cap C_2 = \emptyset$, nên

$$g(\Gamma^+(A)) = g(\Gamma^+(A_1)) - g(C_2) \quad (3).$$

Ngoài ra, $\Gamma^-(v_n) = C_1 \cup C_2$ và $C_1 \cap C_2 = \emptyset$, nên

$$\varphi_{v_n} = \varphi(\Gamma^-(v_n)) = \varphi(C_1) + \varphi(C_2) \quad (4).$$

Từ (1), (2), (3) và (4), ta có:

$$\varphi_{v_n} = \varphi(\Gamma^-(A)) - \varphi(\Gamma^+(A)).$$

5.2.2.5. Định lý (Ford-Fulkerson): Trong mạng vận tải $G=(V,E)$, giá trị lớn nhất của luồng băng thông qua một nhất của thiết diện, nghĩa là

$$\max_{\varphi} \varphi_{v_n} = \min_{A \subset V, v_0 \notin A, v_n \in A} m(\Gamma^-(A)).$$

Chứng minh: Giả sử trong mạng vận tải G , φ^0 là luồng cực đại, mà sau đó bằng phương pháp đánh dấu của thuật toán Ford-Fulkerson không tìm được nữa v_n . Trên cơ sở hiện trạng được đánh dấu của cực đại này, ta dùng B để ký hiệu tập gồm các đỉnh của G không được đánh dấu. Khi đó $v_0 \notin B$, $v_n \in B$. Do đó $\Gamma^-(B)$ là một thiết diện của mạng vận tải G và theo B đề 5.2.2.4, ta có:

$$\varphi_{v_n}^0 = \varphi^0(\Gamma^-(B)) - \varphi^0(\Gamma^+(B)) \quad (1).$$

Đôi với mỗi cung $e=(u,v) \in \Gamma^-(B)$ thì $u \notin B$ và $v \in B$, tức là u được đánh dấu và v không được đánh dấu, nên theo nguyên tắc đánh dấu thì nhất, e đã là cung bão hòa:

$$\varphi^0(e) = m(e).$$

$$\text{Do đó,} \quad \varphi^0(\Gamma^-(B)) = \sum_{e \in \Gamma^-(B)} \varphi^0(e) = \sum_{e \in \Gamma^-(B)} m(e) = m(\Gamma^-(B)) \quad (2).$$

Đôi với mỗi cung $e=(s,t) \in \Gamma^+(B)$ thì $s \in B$ và $t \notin B$, tức là s không được đánh dấu và t được đánh dấu, nên theo nguyên tắc đánh dấu thì hai:

$$\varphi^0(e) = 0.$$

$$\text{Do đó,} \quad \varphi^0(\Gamma^+(B)) = \sum_{e \in \Gamma^+(B)} \varphi^0(e) = 0 \quad (3).$$

Từ (1), (2) và (3) ta suy ra:

$$\varphi_{v_n}^0 = m(\Gamma^-(B)).$$

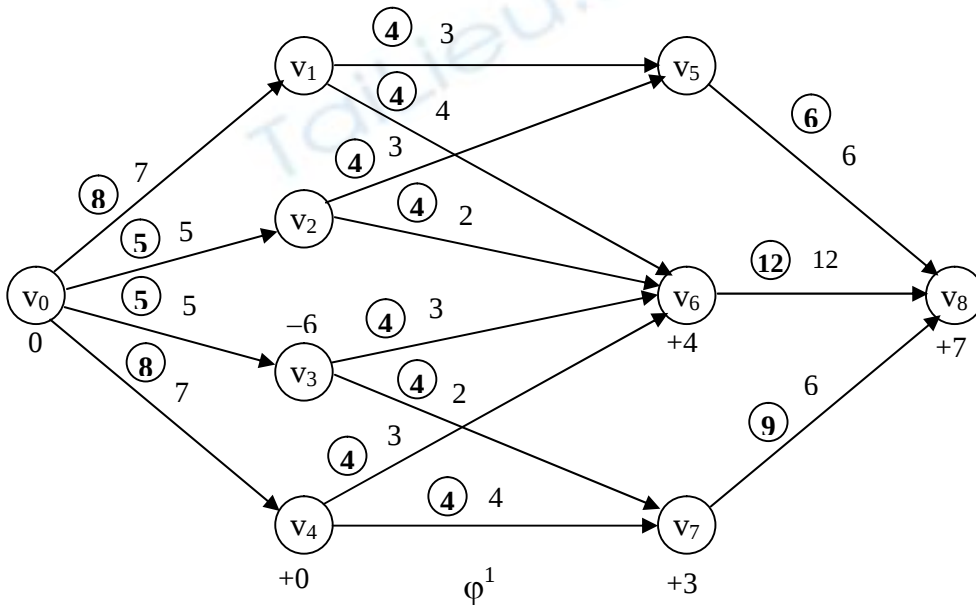
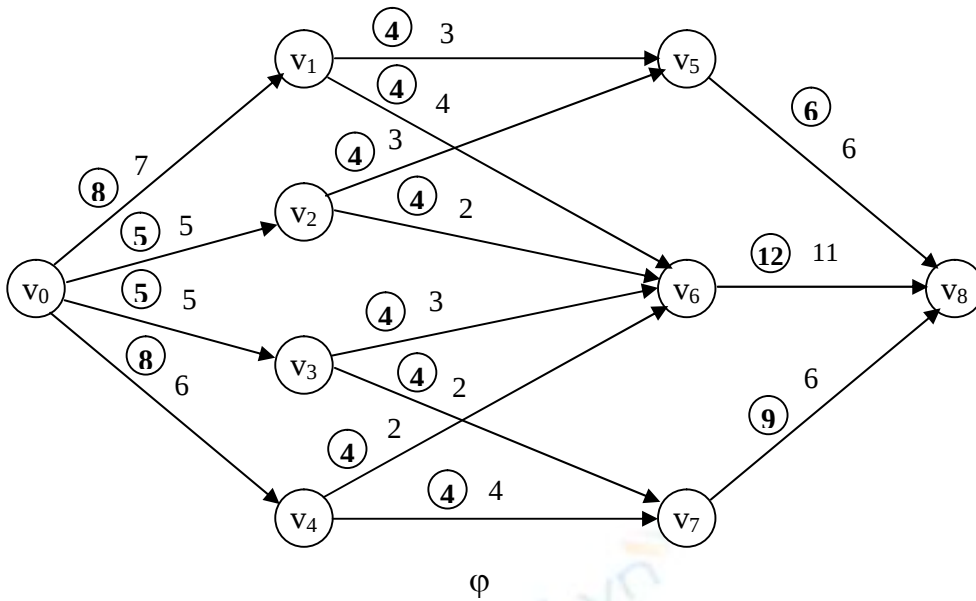
Vì vậy, $\varphi_{v_n}^0$ là giá trị lớn nhất của luồng đi từ v_0 đến v_n , còn $m(\Gamma^-(B))$ là giá trị nhất nhất trong các khả năng thông qua của các thiết diện thu được của mạng vận tải G .

Thí dụ 3: Cho mạng vận tải như hình dưới đây với khả năng thông qua được cho trong khuyên tròn, luồng được ghi trên các cung. Tìm luồng cực đại của mạng này.

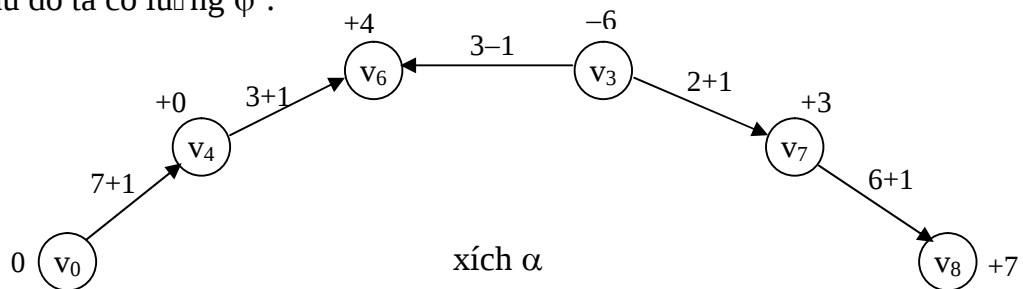
Luồng φ có được đi (v_0, v_4) , (v_4, v_6) , (v_6, v_8) gồm các cung chưa bão hòa nên nó chưa đầy. Do đó có thể nâng luồng của các cung này lên một đơn vị, được φ^1 .

Do mỗi đơn vị xuất phát từ v_0 đến v_8 đều chứa ít nhất một cung bão hòa, nên luồng φ^1 là luồng đầy. Song nó chưa phải là luồng cực đại.

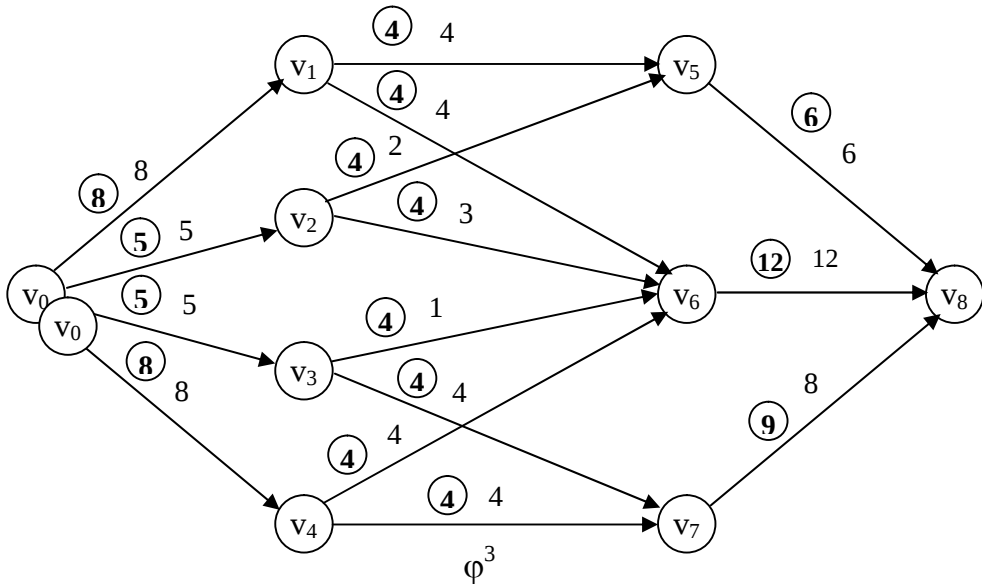
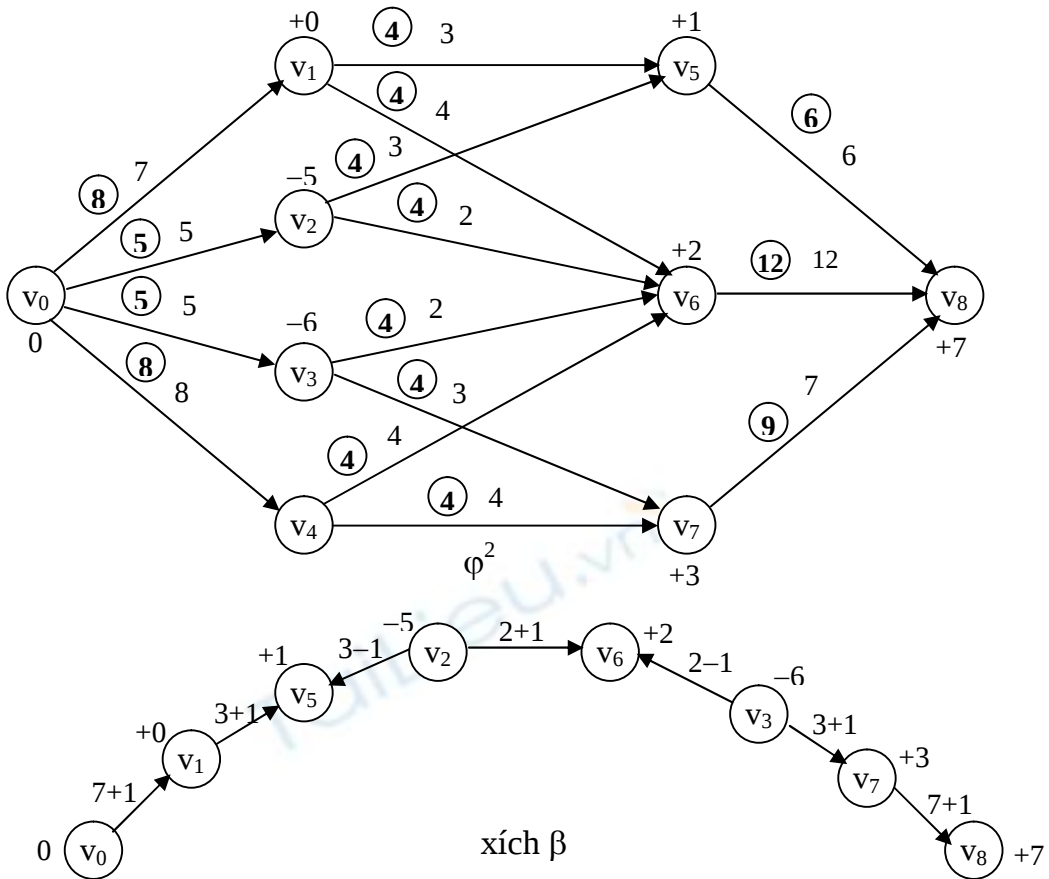
Áp dụng thuật toán Ford-Fulkerson để nâng luồng φ^1 .



Xét xích $\alpha = (v_0, v_4, v_6, v_3, v_7, v_8)$. Quá trình đánh dấu từ v_0 đến v_8 đã có thể nâng luồng φ^1 lên một đơn vị bằng cách biến đổi luồng tại các cung thuộc xích α đã được đánh dấu. Sau đó ta có luồng φ^2 .



Xét xích $\beta = (v_0, v_1, v_5, v_2, v_6, v_3, v_7, v_8)$. Quá trình đánh dấu từ v_0 đến v_8 đã có thể nâng luồng φ^2 lên một đơn vị bằng cách biến đổi luồng tại các cung thuộc xích β đã được đánh dấu. Sau đó ta có luồng φ^3 .



Tiếp theo ta chỉ có thể đánh dấu đến đỉnh v_0 nên quá trình nâng luồng kết thúc và ta dừng giá trị của luồng cực đại là:

$$\phi_{v_8}^3 = 6 + 12 + 8 = 26.$$

Mặt khác, tập đỉnh nhỏ nhất $\Gamma^-(B)$ với $B = \{v_1, v_2, \dots, v_8\}$ là

$$\Gamma^-(B) = \{(v_0, v_1), (v_0, v_2), (v_0, v_3), (v_0, v_4)\}.$$

5.3. BÀI TOÁN DU LỊCH.

5.3.1. Giới thiệu bài toán:

Một người xuất phát từ một thành phố nào đó muốn tìm thăm $n-1$ thành phố khác, mỗi thành phố đúng một lần, rồi quay về thành phố ban đầu. Hỏi nên đi theo trình tự nào để đi dài nhất trong các đồ thị đơn đồ thị đi qua là đồ thị nhúng (kể cả cách gán hai thành phố có thể hiểu là các lý thông thông hoặc thời gian cần đi hoặc chi phí của hành trình, ... và xem như cho trước).

Xét đồ thị đầy đủ $G=(V,E)$, với $V=\{1, 2, \dots, n\}$, có trọng số với trọng số $m_{ij}=m(i,j)$ có thể khác $m_{ji}=m(j,i)$. Như vậy, ta có thể xem G như là một đồ thị có hướng đầy đủ “mạnh” theo nghĩa với mọi $i, j=1, 2, \dots, n, i \neq j$, luôn có $(i,j), (j,i) \in E$. Bài toán trở thành tìm chu trình Hamilton có đi dài nhất trong G .

Bài toán này đã có lời giải bằng cách sử dụng phương pháp “nhánh và cận”.

5.3.2. Phương pháp nhánh và cận: Giả sử trong một tập hợp hữu hạn các phương án của bài toán, ta phải chọn ra một tập phương án tối ưu theo một tiêu chuẩn nào đó (thí dụ làm cho hàm mục tiêu đạt giá trị nhỏ nhất). Ta sẽ tìm cách phân chia tập phương án đang xét thành hai tập con không giao nhau. Với mỗi tập này, ta sẽ tính “cận dưới” (chọn dưới để tối) của các giá trị hàm mục tiêu của các phương án trong đó. Mang so sánh hai cận dưới và tính được, ta có thể phán đoán xem tập con nào có nhiều triển vọng chứa phương án tối ưu và tập tiếp tục phân chia tập con đó thành hai tập con khác không giao nhau, rồi tính các cận dưới tiếp theo ... Loop lại quá trình này thì sau một số hữu hạn bước, cuối cùng sẽ được một tập phương án tối, nói chung là tối ưu. Nếu không thì lặp lại quá trình phân chia để kiểm tra và sau một vài bước, ta sẽ được một tập phương án tối ưu.

Ngay ta thường mô tả quá trình phân chia đó bằng một “cây có gốc” mà gốc sẽ trình bày trình bày cho tập toàn bộ các phương án, còn các đỉnh ở phía dưới là trình bày trình bày cho các tập con trong quá trình “phân nhánh như phân”. Vì vậy, phương pháp này mang tên nhánh và cận.

5.3.3. Các lý luận của phép toán: Nếu không xác định thành phố xuất phát thì có $n!$ hành trình, mỗi hành trình trong với một hoán vị nào đó của tập $\{1, 2, \dots, n\}$. Còn nếu cho trước thành phố xuất phát thì có tất cả là $(n-1)!$ hành trình.

Giả sử $h=(\pi(1), \pi(2), \dots, \pi(n), \pi(1))$ (π là một hoán vị) là một hành trình qua các thành phố $\pi(1), \dots, \pi(n)$ theo thứ tự đó rồi quay về $\pi(1)$ thì hàm mục tiêu

$$f(h) = m_{\pi(1)\pi(2)} + \Lambda + m_{\pi(n-1)\pi(n)} + m_{\pi(n)\pi(1)} = \sum_{(i,j) \in h} m_{ij},$$

sẽ biểu thị tổng đi dài đã đi theo hành trình h , trong đó (i,j) ký hiệu một chặng đi trong của hành trình, tức là một cặp thành phố khác nhau theo hành trình h .

5.3.4. Ma trận rút gọn: Quá trình tính toán số đường đi ngắn nhất trên các ma trận suy từ ma trận trọng số $M=(m_{ij})$ ban đầu bằng những phép biến đổi rút gọn để các số liệu được gọn gàng.

Phép biến đổi như nhất của mỗi dòng (t.đ. c.t) vào tất cả các phần tử của dòng (t.đ. c.t) đó được gọi là phép rút gọn dòng (t.đ. c.t). Phần tử như nhất đó được gọi là hàng số rút gọn dòng (t.đ. c.t) đang xét. Ma trận với các phần tử không âm và có ít nhất một phần tử bằng 0 trên mỗi dòng và mỗi cột được gọi là ma trận rút gọn của ma trận ban đầu.

Thí dụ 4:

$$M = \begin{pmatrix} 4 & 3 & 5 \\ 6 & 2 & 7 \\ 9 & 10 & 5 \end{pmatrix} \begin{matrix} 3 \\ 2 \\ 5 \end{matrix} \longrightarrow \begin{pmatrix} 1 & 0 & 2 \\ 4 & 0 & 5 \\ 4 & 5 & 0 \end{pmatrix} \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \longrightarrow M' = \begin{pmatrix} 0 & 0 & 2 \\ 3 & 0 & 5 \\ 3 & 5 & 0 \end{pmatrix},$$

tất nhiên có thể rút gọn cách khác

$$M = \begin{pmatrix} 4 & 3 & 5 \\ 6 & 2 & 7 \\ 9 & 10 & 5 \end{pmatrix} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \longrightarrow M'' = \begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 2 \\ 5 & 8 & 0 \end{pmatrix}.$$

5.3.5. Mệnh đề: Phép biến đổi ở trên xét trên ma trận trọng số ban đầu cũng là phép biến đổi ở trên của bài toán xét trên ma trận rút gọn và đảo lại.

Chứng minh: Có thể xem việc đi tìm chu trình Hamilton của đồ thị du lịch như là một bài toán với bài đặt biệt đồ thị đang bằng. Như vậy thì trong bằng (ma trận trọng số hoặc ma trận rút gọn) ta phải có đúng n ô chẵn, mỗi ô chẵn tương ứng cho một cặp thành phố trên hành trình cần tìm, trên mỗi dòng và mỗi cột có đúng một ô chẵn. Một hành trình hamiltonian là một chuỗi các cặp thành phố với một tập n ô chẵn xác định. $f(h)$ chính là tổng các trọng số ban đầu ghi trong n ô chẵn đang xét.

Với mỗi hành trình h bất kỳ, nếu ký hiệu $f'(h) = \sum_{(i,j) \in h} m'_{ij}$ là giá trị của hàm mục

tiêu ứng với ma trận rút gọn M' và s là tổng các hàng số rút gọn thì ta có:

$$f(h) = f'(h) + s.$$

Giả sử X là tập toàn bộ các phép biến đổi đang xét ở một giai đoạn nào đó, h_0 là phép biến đổi ở trên của bài toán xét trên ma trận trọng số ban đầu M , ta có:

$$f(h_0) \leq f(h), \forall h \in X$$

hay $f(h_0) - s \leq f(h) - s, \forall h \in X$ hay $f'(h_0) \leq f'(h), \forall h \in X$ hay h_0 là phép biến đổi ở trên của bài toán xét trên ma trận rút gọn M' .

5.3.6. Phân nhánh: Số phân hoạch tập hợp tất cả các hành trình ở một giai đoạn nào đó thành hai tập con rời nhau được biểu diễn bằng số phân nhánh của một cây. Trên cây, mỗi đỉnh được biểu diễn thành một vòng tròn và số tiếp theo trong ngoặc cho một tập hành trình nào đó. Đỉnh X đầu tiên là tập toàn bộ các hành trình. Đỉnh (i, j) biểu diễn tập các hành trình có chứa cặp (i, j) khác nhau. Đỉnh (i, j) biểu diễn tập các hành trình không chứa cặp (i, j) khác nhau. Từ đỉnh (i, j) lại có số phân nhánh: đỉnh (k, l) biểu diễn tập các hành trình có chứa cặp (i, j) và cặp (k, l) , đỉnh (k, l) biểu diễn tập các hành trình có chứa cặp (i, j) nhưng không chứa cặp (k, l) ...

Nếu quá trình diễn ra đến lúc thì cuối cùng sẽ có những đỉnh chỉ biểu diễn một hành trình duy nhất.

Vấn đề đặt ra là nên chọn cặp thành phần nào để tiến hành phân nhánh xuất phát từ một đỉnh cho trước trên cây? Một cách tự nhiên ta nên chọn cặp thành phần nào gần nhau nhất để phân nhánh trước, trên ma trận rút gọn thì những cặp thành phần (i, j) như vậy đều có $m'_{ij}=0$ và những hành trình nào chứa cặp (i, j) đều có trị số vòng là tối.

Trên ma trận rút gọn thông thường có nhiều cặp thành phần thỏa mãn điều kiện đó ($m'_{ij}=0$). Để quyết định ta phải tìm cách so sánh. Vì thành phần i nhất thiết phải nằm liên với một thành phần nào đó nên các hành trình h không chứa (i, j) tức là $h \in \overline{(i, j)}$ phải không những đủ dài hành trình ít ra có chứa phần tử nhỏ nhất trong dòng i không có $m'_{ij}=0$ và phần tử nhỏ nhất trong cột j không có $m'_{ij}=0$ vì thành phần j nhất thiết phải nằm liên với một thành phần nào đó ở trước nó trên hành trình. Ký hiệu tổng của hai phần tử nhỏ nhất đó là θ_{ij} thì ta có $f'(h) \geq \theta_{ij}, \forall h \in \overline{(i, j)}$.

Vì lý do trên, số θ_{ij} có thể dùng làm tiêu chuẩn so sánh giữa các cặp thành phần (i, j) cùng có $m'_{ij}=0$. Một cách tổng quát, ở mỗi giai đoạn ta sẽ chọn cặp thành phần (i, j) có $m'_{ij}=0$ trong ma trận rút gọn và có θ_{ij} lớn nhất để tiến hành phân nhánh từ một đỉnh trên cây.

5.3.7. Tính cận: Với mỗi đỉnh của cây phân nhánh, ta phải tính cận dưới của các giá trị hàm mục tiêu trong tập hợp phần tử án mà đỉnh đó biểu diễn. Cận dưới tính được sẽ ghi bên dưới đỉnh đang xét.

Theo công thức $f(h)=f'(h)+s$ và do $f'(h) \geq 0$ nên $f(h) \geq s, \forall h \in X$. Vì vậy tổng các hàng sẽ rút gọn của ma trận ban đầu có thể lấy làm cận dưới của đỉnh X đầu tiên trên cây. Một khác, ta lại có $f'(h) \geq \theta_{ij}, \forall h \in \overline{(i, j)}$, do đó $f(h)=f'(h)+s \geq \theta_{ij}+s, \forall h \in \overline{(i, j)}$. Vì vậy tổng $\theta_{ij}+s$ có thể lấy làm cận dưới cho đỉnh (i, j) . Sau khi chọn (i, j) để phân nhánh xuất phát từ đỉnh X thì trên bảng có thể xóa dòng i và cột j vì trên đó ô chứa (i, j) là duy nhất. Sau khi bỏ dòng i và cột j thì ma trận M' lại có thể rút gọn thành ma trận M'' với s' là tổng các hàng sẽ rút gọn, $f''(h)$ là giá trị của hàm mục tiêu xét trên M'' . Khi đó ta có $f'(h)=f''(h)+s', \forall h \in (i, j)$, do đó $f(h)=f'(h)+s=f''(h)+s+s', \forall h \in (i, j)$. Do $f''(h) \geq 0$ nên

$f(h) \geq s+s', \forall h \in (i,j)$, nghĩa là tổng $s+s'$ có thể lấy làm cận dưới cho đỉnh (i,j) trong cây phân nhánh.

Nếu tiếp tục phân nhánh thì cận dưới của các đỉnh tiếp sau được tính toán tương tự, vì đây là một quá trình lặp. Ta chỉ cần xem đỉnh xuất phát của các nhánh giống như đỉnh X ban đầu.. Để tiết kiệm khi lập trình tính toán, ngược lại ta thường chỉ cần đỉnh có cận dưới nhỏ nhất để phân nhánh tiếp tục.

5.3.8. Thuật toán ngăn chặn hành trình con: Một đường đi hoặc chu trình Hamilton không thể chứa chu trình con với số cạnh tạo thành nhỏ hơn n . Vì vậy ta sẽ đặt $m_{ij} = \infty$ ($i=1, \dots, n$) để tránh các khuyết.

Với $i \neq j$ và nếu (i,j) là ô chẵn thì phải đặt ngay $m'_{ji} = \infty$ trong ma trận rút gọn.

Nếu đã chẵn (i,j) và (j,k) và $n > 3$ thì phải đặt ngay $m'_{ji} = m'_{kj} = m'_{ki} = \infty$.

Chú ý rằng việc đặt $m'_{ij} = \infty$ tương đương với việc xoá ô (i,j) trong bảng hoặc xem (i,j) là ô cấm, nghĩa là hai thành phần i và j không được kết nhau trong hành trình đơn liên liên tiếp. Ở mỗi giai đoạn của quá trình đều phải tiến hành thuật toán ngăn chặn này trước khi tiếp tục rút gọn ma trận.

5.3.9. Tính chất tối ưu: Quá trình phân nhánh, tính cận, ngăn chặn hành trình con, rút gọn ma trận phải thực hiện cho đến khi nào có được ô chẵn để liên tiếp một hành trình Hamilton, nói cách khác là cho đến khi trên cây phân nhánh đã xuất hiện một đỉnh chỉ biểu diễn một hành trình duy nhất và đã xoá hết được mọi vòng lặp trong bảng. Cận dưới của đỉnh cuối cùng này chính là độ dài của hành trình yêu cầu liên tiếp.

a) Nếu cận dưới của đỉnh này không lớn hơn các cận dưới của mọi đỉnh treo trên cây phân nhánh thì hành trình đó là tối ưu.

b) Nếu trái lại thì phải xuất phát từ đỉnh treo nào có cận dưới nhỏ hơn để phân nhánh tiếp tục và kiểm tra xem liệu khi nào a) có thỏa mãn không.

Thí dụ 5: Xét bài toán với 6 thành phần, các số liệu cho theo bảng sau:

$$M = \begin{pmatrix} \infty & 27 & 43 & 16 & 30 & 26 \\ 7 & \infty & 14 & 1 & 30 & 25 \\ 20 & 13 & \infty & 35 & 5 & 0 \\ 21 & 16 & 25 & \infty & 18 & 18 \\ 12 & 46 & 27 & 48 & \infty & 5 \\ 23 & 5 & 5 & 9 & 5 & \infty \end{pmatrix} \begin{matrix} 16 \\ 1 \\ 0 \\ 16 \\ 5 \\ 5 \end{matrix}$$

$$\begin{matrix} 5 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Tổng các hàng sẽ rút gọn bằng cách đầu là $s=48$. Trong ma trận rút gọn ta có:

$$m'_{14} = m'_{24} = m'_{36} = m'_{41} = m'_{42} = m'_{56} = m'_{62} = m'_{63} = m'_{65} = 0$$

và $\theta_{14}=10$, $\theta_{24}=1$, $\theta_{36}=5$, $\theta_{41}=1$, $\theta_{42}=0$, $\theta_{56}=2$, $\theta_{62}=0$, $\theta_{63}=9$, $\theta_{65}=2$. Sau khi so sánh ta thấy $\theta_{14}=10$ là lớn nhất nên ta chọn ô (1,4) để phân nhánh. Chọn dòng i của đỉnh $\overline{(1,4)}$ là $s+\theta_{14}=58$. Xóa dòng 1 cột 4 rồi đặt $m'_{41}=\infty$.

$$M' = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} \infty & 11 & 27 & 0 & 14 & 10 \\ 1 & \infty & 13 & 0 & 29 & 24 \\ 15 & 13 & \infty & 35 & 5 & 0 \\ 0 & 0 & 9 & \infty & 2 & 2 \\ 2 & 41 & 22 & 43 & \infty & 0 \\ 13 & 0 & 0 & 4 & 0 & \infty \end{pmatrix} \end{matrix}.$$

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 5 & 6 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 1 & \infty & 13 & 29 & 24 \\ 15 & 13 & \infty & 5 & 0 \\ \infty & 0 & 9 & 2 & 2 \\ 2 & 41 & 22 & \infty & 0 \\ 13 & 0 & 0 & 0 & \infty \end{pmatrix} \end{matrix} \longrightarrow M'' = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 5 & 6 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & \infty & 12 & 28 & 23 \\ 15 & 13 & \infty & 5 & 0 \\ \infty & 0 & 9 & 2 & 2 \\ 2 & 41 & 22 & \infty & 0 \\ 13 & 0 & 0 & 0 & \infty \end{pmatrix} \end{matrix}.$$

Tổng hàng số rút gọn là $s'=1$. Với cột i của đỉnh (1,4) là $s+s'=49$. Vì $49 < 58$ nên tiếp tục phân nhánh từ đỉnh (1,4). Trong ma trận còn lại, sau khi rút gọn ta có

$$m''_{21}=m''_{36}=m''_{42}=m''_{56}=m''_{62}=m''_{63}=m''_{65}=0.$$

Ở giai đoạn này, sau khi tính toán ta thấy $\theta_{21}=14$ là lớn nhất nên chọn tiếp ô (2,1). Chọn dòng i của đỉnh $\overline{(2,1)}$ là $49+\theta_{21}=63$. Xóa dòng 2 cột 1. Đặt $m''_{42}=\infty$. Rút gọn ma trận còn lại, ta có:

$$\begin{matrix} & \begin{matrix} 2 & 3 & 5 & 6 \end{matrix} \\ \begin{matrix} 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 13 & \infty & 5 & 0 \\ \infty & 9 & 2 & 2 \\ 41 & 22 & \infty & 0 \\ 0 & 0 & 0 & \infty \end{pmatrix} \end{matrix} \longrightarrow M''' = \begin{matrix} & \begin{matrix} 2 & 3 & 5 & 6 \end{matrix} \\ \begin{matrix} 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 13 & \infty & 5 & 0 \\ \infty & 7 & 0 & 0 \\ 41 & 22 & \infty & 0 \\ 0 & 0 & 0 & \infty \end{pmatrix} \end{matrix}.$$

Tổng hàng số rút gọn là 2. Chọn dòng i của đỉnh (2,1) là $49+2=51$.

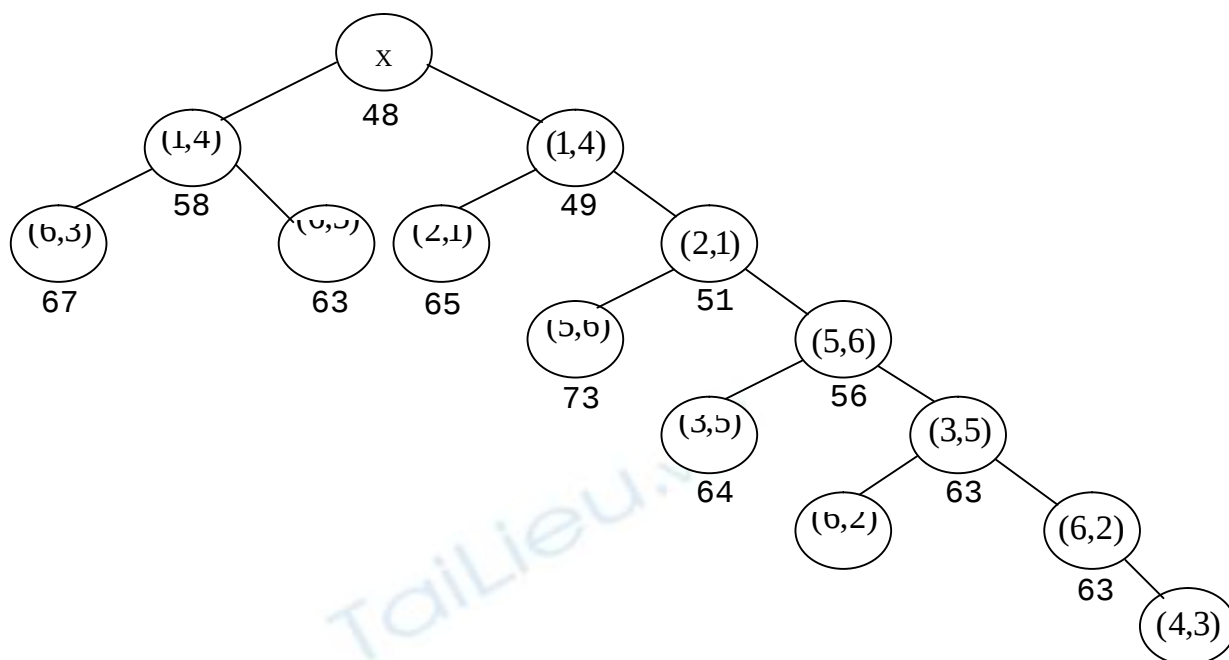
Tiếp tục như vậy cuối cùng ta được 6 ô chọn là:

$$(1,4), (2,1), (5,6), (3,5), (4,3), (6,2)$$

và kiểm tra hành trình $h_0=(1\ 4\ 3\ 5\ 6\ 2\ 1)$ với $f(h_0)=63$ là chọn dòng i của đỉnh cuối cùng. Chọn dòng i của đỉnh cuối cùng là 63, trong khi đó đỉnh treo $\overline{(1,4)}$ có chọn dòng i là $58 < 63$ nên phải tiếp tục phân nhánh từ đó để kiểm tra. Sau số phân nhánh này thì mọi đỉnh treo đều có chọn dòng i không nhỏ hơn 63 nên có thể khẳng định rằng hành trình $h_0=(1\ 4\ 3\ 5\ 6\ 2\ 1)$ là tối ưu.

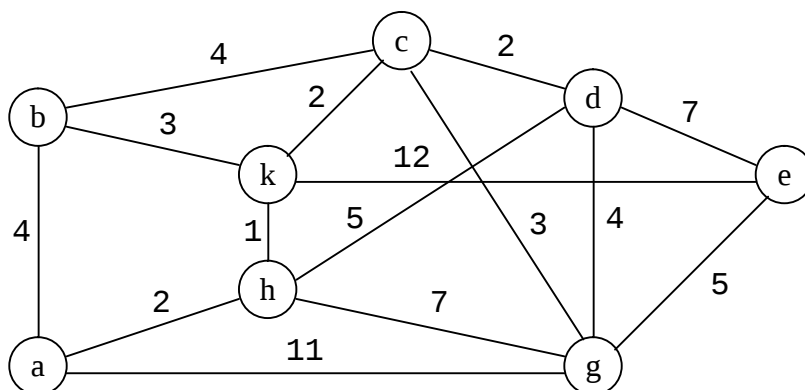
Số phân nhánh từ đỉnh $\overline{(1,4)}$ để làm như sau: trong ma trận rút gọn đợt 1, ta đặt $m'_{14}=\infty$ vì xem ô (1,4) là ô cấm, $\theta_{63}=9$ là lớn nhất trong các θ_{ij} , do đó chọn ô (6,3) để

phân nhánh. Công dồn i của đỉnh $\overline{(6,3)}$ là $58+9=67$. Đợt $m'_{36}=\infty$. Rút gọn ma trận với tổng hàng số rút gọn là 15. Công dồn i của đỉnh $(6,3)$ là $58+15=73$.

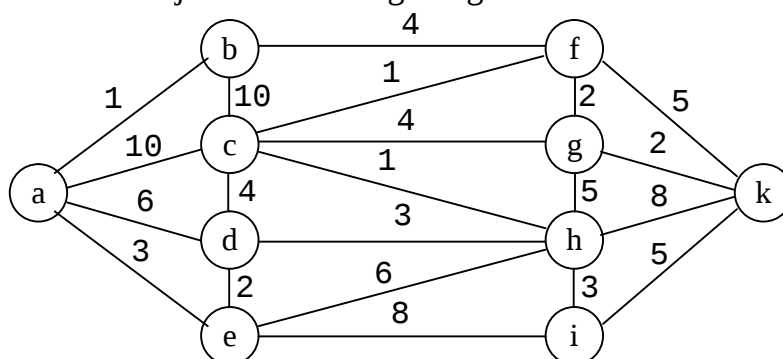


BÀI TẬP CHỌN NG V:

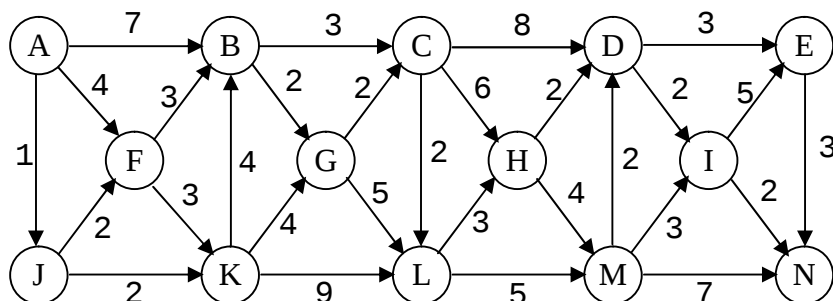
1. Dùng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh a đến các đỉnh khác trong đồ thị sau:



2. Dùng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh a đến các đỉnh khác trong đồ thị sau:



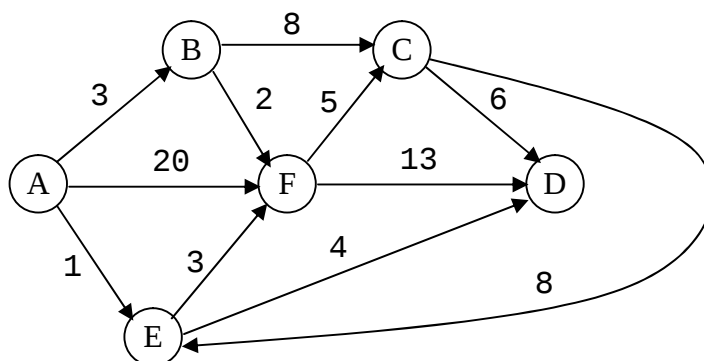
3. Cho đồ thị có trọng số như hình dưới đây. Hãy tìm đường đi ngắn nhất từ đỉnh A đến đỉnh N.



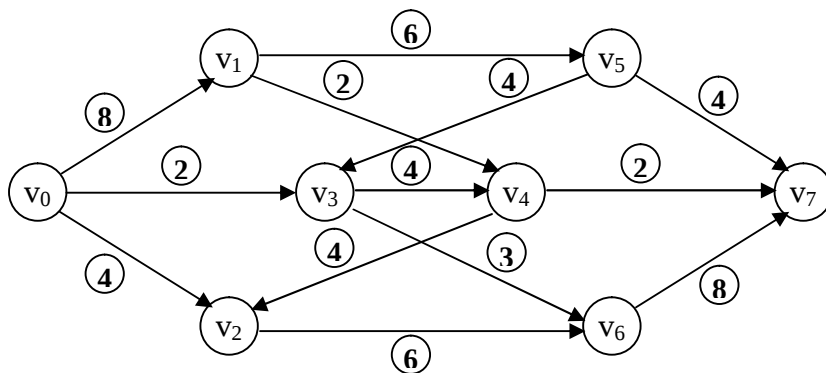
4. Tìm đường đi ngắn nhất từ B đến các đỉnh khác của đồ thị có ma trận trọng số là (các ô trống là ∞):

	A	B	C	D	E	F	G
A		3	6				
B	3		2	4			
C	6	2		1	4	2	
D		4	1		2		4
E			4	2		2	1
F			2		2		4
G				4	1	4	

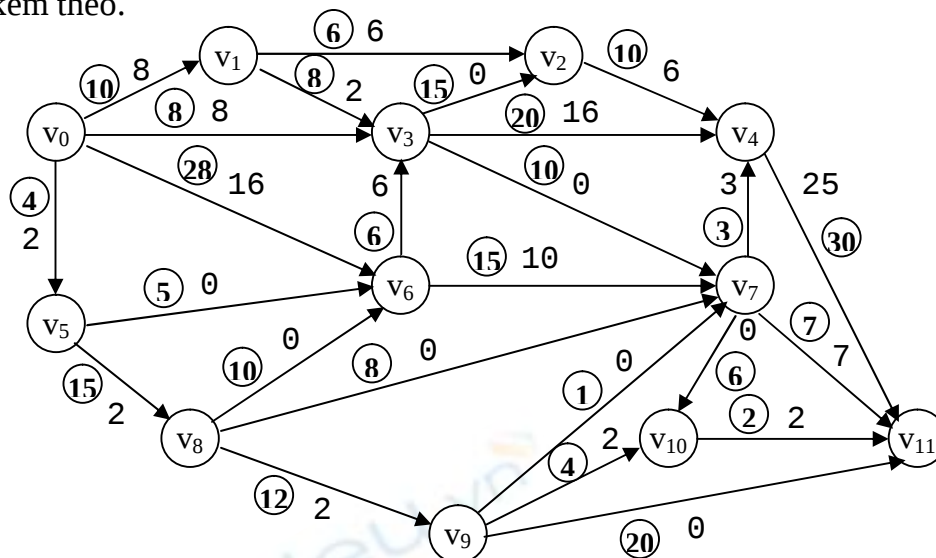
5. Tìm W^* bằng cách áp dụng thuật toán Floyd vào đồ thị sau:



6. Giải bài toán mạng vận tải sau bằng thuật toán Ford-Fulkerson với luồng vận tải khởi đầu bằng 0.



7. Giải bài toán mạng vận tải sau bằng thuật toán Ford-Fulkerson với lưu lượng vận tải khởi đầu bằng 0 kèm theo.



8. Hãy giải bài toán vận tải với 6 thành phố, có số liệu cho trong ma trận trình bày sau:

$$\begin{pmatrix} \infty & 25 & 45 & 14 & 32 & 24 \\ 9 & \infty & 16 & 2 & 34 & 23 \\ 22 & 11 & \infty & 33 & 7 & 0 \\ 23 & 14 & 27 & \infty & 20 & 21 \\ 14 & 44 & 29 & 46 & \infty & 3 \\ 25 & 3 & 4 & 7 & 8 & \infty \end{pmatrix}.$$