MẠNG VÀ HÀM TẢI

GV: Lê Mậu Long

Nội dung

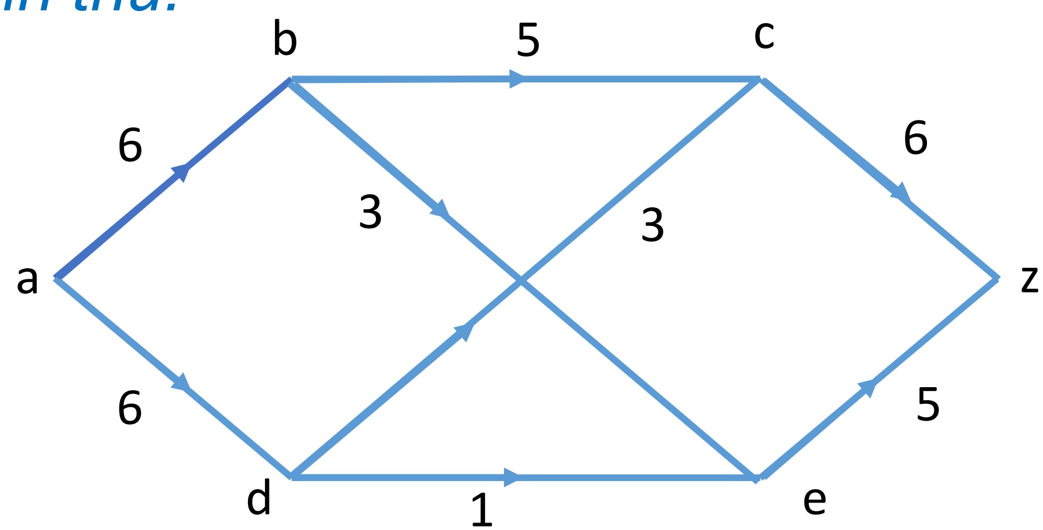
- ✓ Mạng và hàm tải.
- ✓ Hàm tải tối đại – Thuật toán FORD-FULKERSON.

I. Mạng và hàm tải

Định nghĩa

Mạng (network) là đơn đồ thị có hướng có trọng số $G = (V, E)$, trên đó đã chọn:

- Đỉnh a gọi là *đỉnh phát,*
- Đỉnh z gọi là *đỉnh thu.*



Định nghĩa

Xét mạng $G = (V, E)$, với mỗi đỉnh x, đặt:

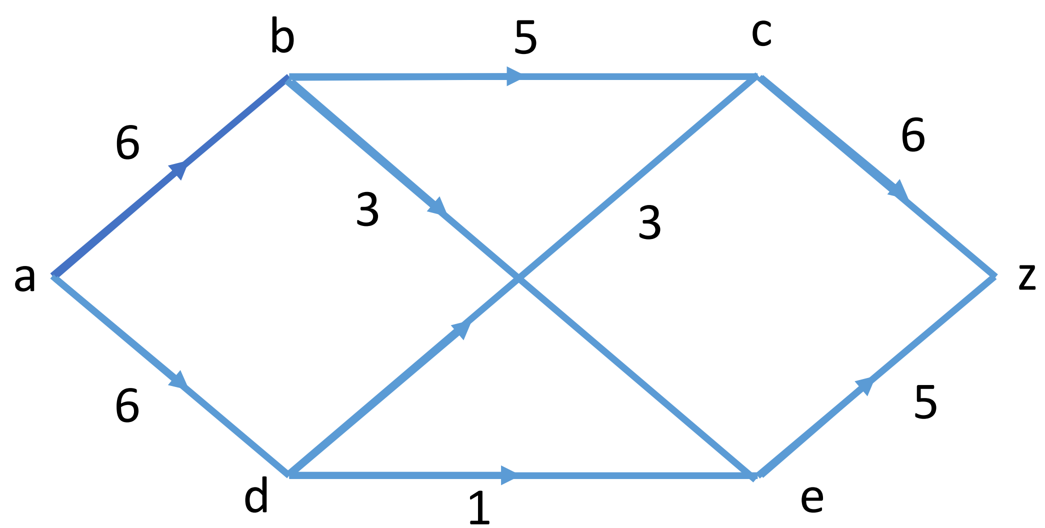
$$In(x) = \{e \in E \mid e \text{ tới trong } x\}$$

$$Out(x) = \{e \in E \mid e \text{ tới ngoài } x\}$$

Ví dụ:

$In(b) = \{\overrightarrow{ab}\}$

$Out(b) = \{\overrightarrow{bc},\ \overrightarrow{be}\}$



Định nghĩa

Một *hàm tải* (flow function) trên mạng G là một hàm

$$\varphi : E \to N$$

Thỏa các điều kiện

1. $0 \le \varphi(e) \le c(e), \qquad \forall e \in E$
2. $\varphi(e) = 0, \qquad \forall e \in In(a) \cup Out(z)$
3. $\sum_{e \in In(x)} \varphi(e) = \sum_{e \in Out(x)} \varphi(e)\ , \forall x \in V \backslash \{a, z\}$

Trong đó $c(e)$ là trọng số cạnh e

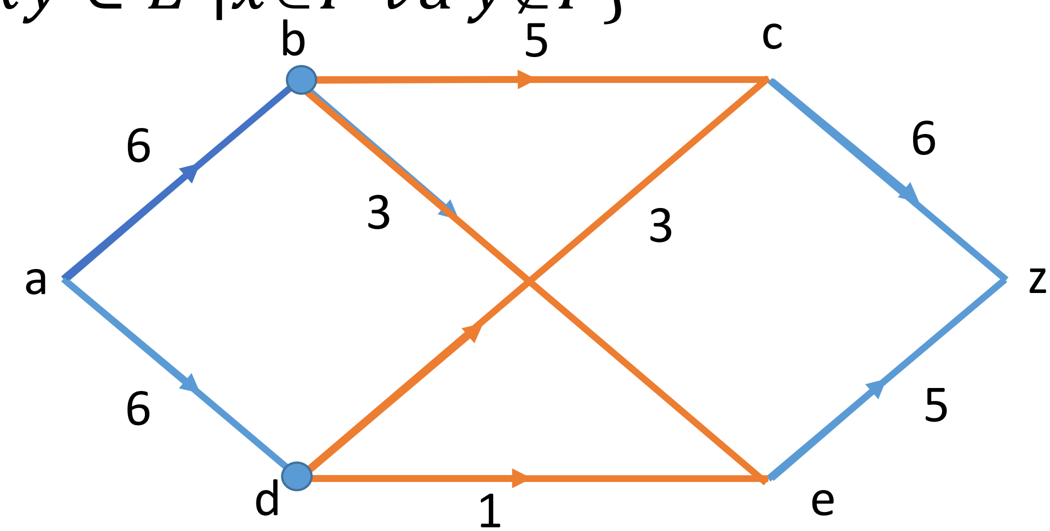
Định nghĩa

Phép cắt: Xét $P \subset V$, phép cắt của P ký hiệu $(P, \bar{P})$ là tập hợp:

$$(P, \bar{P}) = \{e = \overrightarrow{xy} \in E \mid x \in P \text{ và } y \notin P\}$$

Ví dụ: $P = \{b, d\}$

$$(P, \bar{P}) = \{\overrightarrow{bc}, \overrightarrow{be}, \overrightarrow{dc}, \overrightarrow{de}\}$$



Nếu $a \in P$ và $z \notin P$ thì $(P, \bar{P})$ được gọi là *phép cắt a-z*

Định lý

Cho φ là hàm tải trên mạng G, $P \subset V\backslash\{a, z\}$ thì

$$\sum_{e\in(\overline{P},P)} \varphi(e) = \sum_{e\in(P,\overline{P})} \varphi(e)$$

CM: Theo tính chất *3*. của hàm tải ta có:

$$\sum_{e\in In(x)} \varphi(e) = \sum_{e\in Out(x)} \varphi(e)$$

Lấy tổng 2 vế theo $x \in P$

$$\sum_{x\in P}\sum_{e\in In(x)} \varphi(e) = \sum_{x\in P}\sum_{e\in Out(x)} \varphi(e)$$

Đơn giản 2 vế cho những cạnh $\overrightarrow{xy}$ với $x, y \in P$, suy ra điều phải chứng minh

Định lý

Hệ quả: *Cho φ là hàm tải trên mạng G, thì*

$$\sum_{e \in Out(a)} \varphi(e) = \sum_{e \in In(z)} \varphi(e)$$

CM: Áp dụng định lý với $P = V \backslash \{a, z\}$

$$\sum_{e \in Out(a)} \varphi(e) = \sum_{e \in (\overline{P}, P)} \varphi(e) = \sum_{e \in (P, \overline{P})} \varphi(e) = \sum_{e \in In(z)} \varphi(e)$$

Tải trọng hàm tải

- *Tải trọng hàm tải* φ, ký hiệu $|\varphi|$ được định nghĩa như sau:

$$|\varphi| = \sum_{e \in Out(a)} \varphi(e) = \sum_{e \in In(z)} \varphi(e)$$

- *Trọng số phép cắt* $(P, \overline{P})$ được định nghĩa như sau:

$$C(P, \overline{P}) = \sum_{e \in (P, \overline{P})} c(e)$$

Bài toán đặt ra là tìm hàm tải tối đại (tải trong lớn nhất) trên mạng G

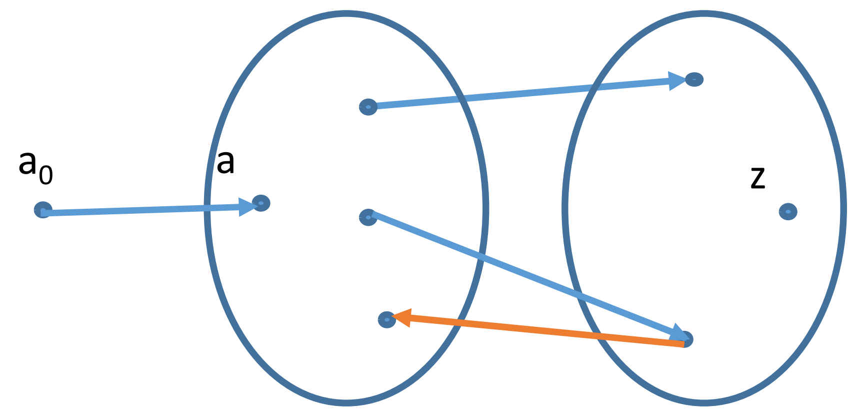
Định lý

φ là hàm tải trên mạng G, $(P,\overline{P})$ là phép cắt a-z, thì

$$|\varphi| \le c(P,\overline{P})$$

CM: Từ mạng G ta xây dựng mạng G' bằng cách thêm cạnh $\overrightarrow{a_0a}$ với trọng số là ∞ và hàm φ'

$$\varphi'(e) = \begin{cases} |\varphi| & \text{với } e = \overrightarrow{a_0a} \\ \varphi(e) & e \ne \overrightarrow{a_0a} \end{cases}$$



Đặt $P' = P \Rightarrow \overline{P'} = \overline{P} \cup \{a_0\}$

$$|\varphi| = |\varphi'| = \sum_{e\in Out(a_0)} \varphi'(e) \le \sum_{e\in(\overline{P'},P')} \varphi'^{(e)} = \sum_{e\in(P',\overline{P'})} \varphi'^{(e)} = \sum_{e\in(P,\overline{P})} \varphi(e) \le \sum_{e\in(P,\overline{P})} c(e) = C(P,\overline{P})$$

Hệ quả

Từ chứng minh định lý, suy ra hệ quả sau:

φ là hàm tải trên mạng G, $(P, \overline{P})$ là phép cắt a-z, thì

$$|\varphi| = c(P, \overline{P}) \Leftrightarrow \begin{cases} \varphi(e) = 0; & \forall e \in (\overline{P}, P) \\ \varphi(e) = c(e); & \forall e \in (P, \overline{P}) \end{cases}$$

II. Thuật toán Ford-Fulkerson

Thuật toán FORD-FULKERSON

- *Dây chuyền K* là đường vô hướng từ $a\ đến\ z$
- *Độ lệch tải*: Với $e \in E$, độ lệch tải trên e:

$$s(e) = c(e) - \varphi(e)$$

Nếu $s(e) = 0$, ta nói φ bảo hòa trên e

- Xét dây chuyền K, ta định nghĩa hàm

$$\varphi_K : E \rightarrow \{-1, 0, 1\}$$

Với

$$\varphi_{K(e)} = \begin{cases} 1 \quad nếu\ e \in K, hướng\ từ\ a\ đến\ z \\ 0 \qquad\qquad\quad nếu\ e \notin \in K \\ -1\ nếu\ e \in K, hướng\ từ\ z\ đến\ a \end{cases}$$

Thuật toán FORD-FULKERSON

Khởi đầu từ hàm tải φ bất kỳ trên mạng G

B1: Mọi đỉnh chưa có nhãn và chưa xét, gán nhãn cho a là (-, ∞). Đặt $p = a$.

B2: Xét p, gán nhãn cho các đỉnh q chưa có nhãn kề p bằng thủ tục sau:

- Với $\mathrm{e} = \overrightarrow{pq}, và\ s(e) > 0$, gán nhãn cho q là $(p^+, \Delta q)$ với $\Delta q = \min(s(e), \Delta p)$
- Với $\mathrm{e} = \overrightarrow{qp}, và\ \varphi(e) > 0$, gán nhãn cho q là $(p^-, \Delta q)$ với $\Delta q = \min(\varphi(e), \Delta p)$

Thuật toán FORD-FULKERSON

B3: Nếu z có nhãn, lần ngược theo thành phần thứ nhất của nhãn xác định dây chuyền K và cập nhật lại hàm φ:

$$\varphi(e) = \varphi(e) + \varphi_K(e).\Delta z$$

Quay về B1.

B4: Tìm 1 đỉnh có nhãn nhưng chưa xét

- Nếu tìm thấy, xem đỉnh này là p → quay về B2
- Nếu không tìm thấy → Dừng thuật toán.

Khi thuật toán dừng P là tập hợp các đỉnh có nhãn tạo thành phép cắt $(P, \overline{P})$ tối tiểu qua đó hàm tải tối đại

Ví dụ

Xét mạng G và hàm tải φ như hình vẽ

Gán nhãn cho a là (-, ∞)

Xét a

 Gán nhãn cho d là $(a^+, 4)$

Xét d

 Gán nhãn cho c là $(d^+, 2)$

Xét c

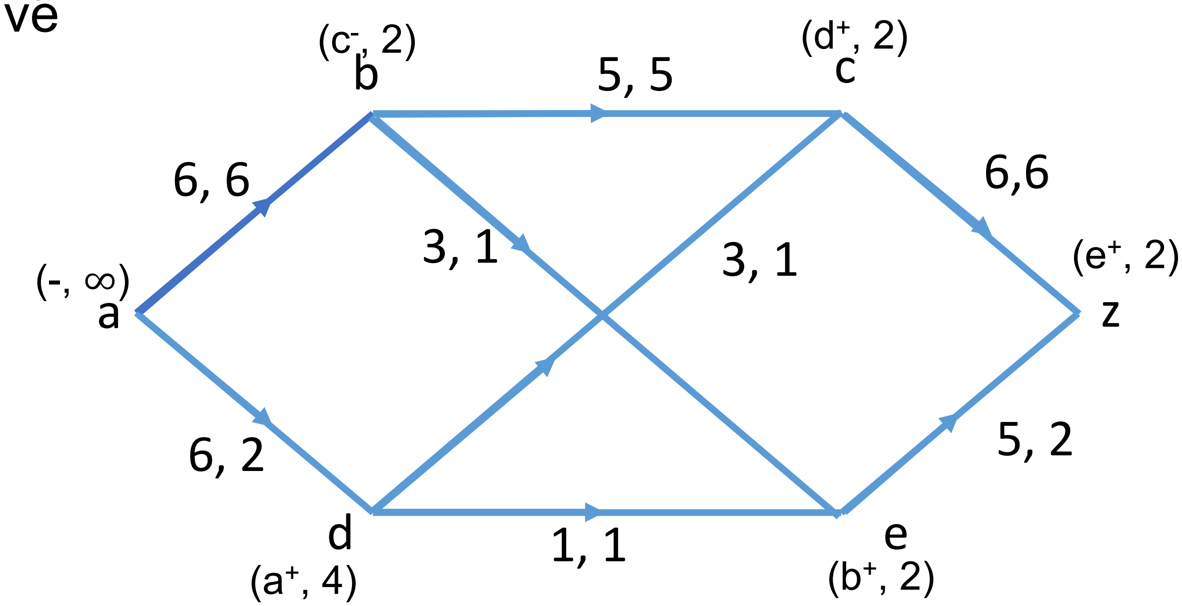
 Gán nhãn cho b là $(c^-, 2)$

Xét b

 Gán nhãn cho e là $(b^+, 2)$

Xét e

 Gán nhãn cho z là $(e^+, 2)$



$$|\varphi| = 6 + 2 = 8$$

Ví dụ

Z đã có nhãn, lần ngược theo thành phần thứ nhất của nhãn, xác định được dây chuyền K:

$$a \to d \to c \leftarrow b \to e \to z$$

Cập nhật lại hàm φ

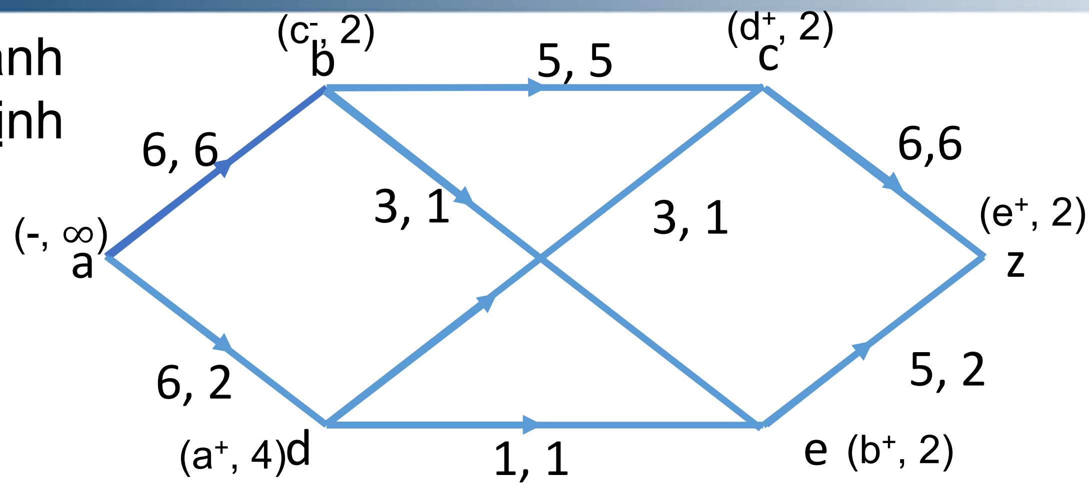
$$\varphi(\overrightarrow{ez}) = 2 + 2 = 4$$

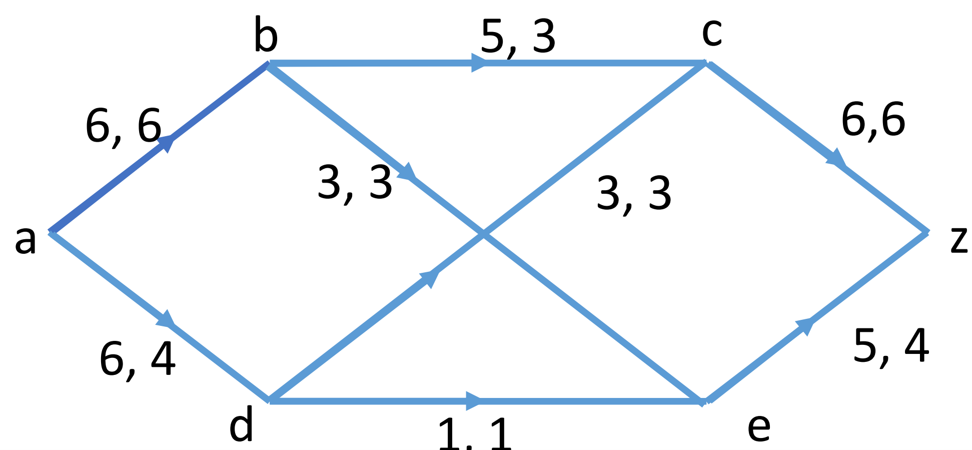
$$\varphi\left(\overrightarrow{be}\right) = 1 + 2 = 3$$

$$\varphi\left(\overrightarrow{bc}\right) = 5 - 2 = 3$$

$$\varphi\left(\overrightarrow{dc}\right) = 1 + 2 = 3$$

$$\varphi\left(\overrightarrow{ad}\right) = 2 + 2 = 4$$





Chu trình Euler

Gán nhãn cho a là (-, ∞)

Xét a

Gán nhãn cho d là (a^+, 2)

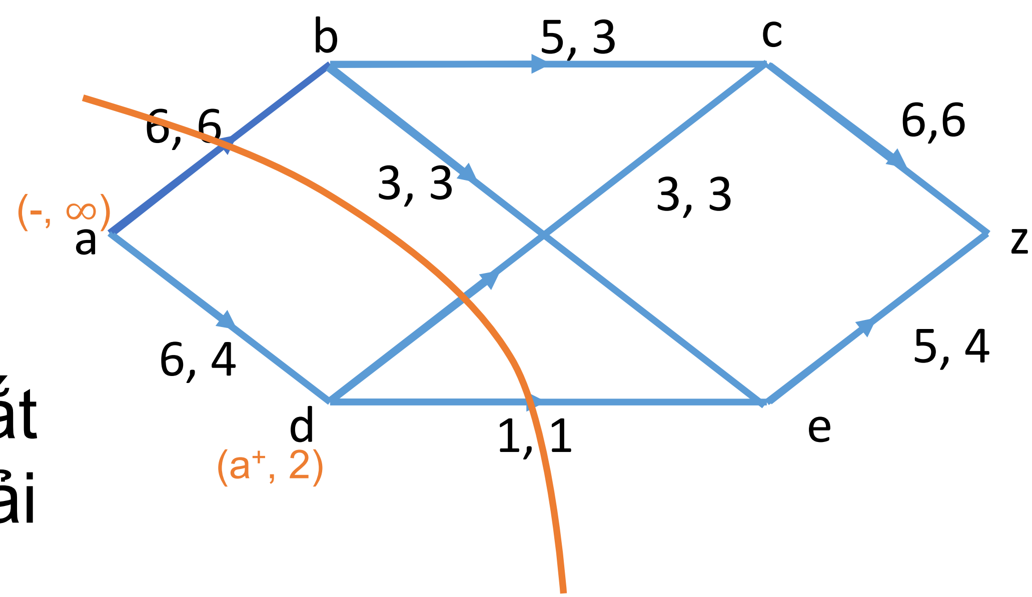
Xét d

Không gán nhãn được

Thuật toán dừng

$P = \{a, d\}$ tạo thành phép cắt $(P, \bar{P})$ tối tiểu và qua đó hàm tải tối đại

$|\varphi| = 8$



Tóm tắt

- ✓ Mạng và hàm tải.
- ✓ Hàm tải tối đại – Thuật toán FORD-FULKERSON.