05. LUỒNG TRONG MẠNG

Bài giảng Lý thuyết đồ thị

Đặng Nguyễn Đức Tiến

HCMUS – 2009

Nội dung

- Giới thiệu
- Luồng trong mạng
- Bài toán luồng cực đại
- Thuật toán Ford Fulkerson
- Một số ứng dụng của bài toán luồng cực đại

Tài liệu tham khảo

- Nguyễn Đức Nghĩa, Nguyễn Tô Thành, **Toán rời rạc**, ltb. 1, nxb. Giáo dục, 1998, ch. 7, tr. 215 – 236.
- Đỗ Minh Hoàng, **Bài giảng Chuyên đề Giải thuật & Lập trình**, ĐHSP Hà Nội, 2004, ch. 10, tr. 257 – 267.
- Dương Anh Đức, Trần Đan Thư, **Bài giảng lý thuyết đồ thị**, 2002, ch. 5.

Giới thiệu

- Luồng cực đại là một trong những bài toán tối ưu trên đồ thị tìm được những ứng dụng rất rộng rãi trong cả thực tế cũng như trong lý thuyết tổ hợp.
- Bài toán được đề xuất vào đầu những năm 1950 và gắn liền với tên tuổi của 2 nhà toán học Mỹ: **Ford** (Lester Randolph Ford: 1927 -) và **Fulkerson** (Delbert Ray Fulkerson: 1924 - 1976).

Các định nghĩa – Mạng

- *Mạng (network)* là một đồ thị có hướng $G = (V, E)$ trong đó:
  - Có duy nhất một đỉnh *s* không có cung đi vào, được gọi là đỉnh phát (source)
  - Có duy nhất một đỉnh t không có cung đi ra, được gọi là đỉnh thu (sink)
  - Mỗi cạnh *e* = (u, v) $\in$ E được gán một số nguyên không âm *c(e)* = c[u, v] và gọi là *khả năng thông qua* của cung đó (capacity).
- Ta quy ước nếu mạng không có cung (u, v) thì ta thêm vào cung (u, v) với khả năng thông qua c[u, v] bằng 0.

Các định nghĩa – Tập cung vào ra

- Với một mạng G = (V, E, c), ta ký hiệu:
  - $W^-(x) = \{(u, v) \in E \mid u \in V\}$: tập các cung đi vào đỉnh v.
  - $W^+(x) = \{(v, u) \in E \mid u \in V\}$: tập các cung đi ra khỏi đỉnh v.

Các định nghĩa – Luồng trên mạng

- Giả sử cho mạng G = (V, E). Ta gọi luồng f trong mạng là ánh xạ f: E → R_+ gán cho mỗi cung e = (u, v) $\in$ E một số thực không âm f(e) = f[u, v], thoả mãn các điều kiện sau:
  - ĐK 1 (Capacity Constraint): Luồng trên mỗi cung e $\in$ E không vượt quá khả năng thông qua của nó:

    $$0 \le f(e) \le c(e)$$

  - ĐK 2 (Flow Conversion): Điều kiện cân bằng luồng trên mỗi đỉnh của mạng: Tổng luồng trên các cung vào đỉnh v bằng tổng luồng trên các cung đi ra khỏi đỉnh v, nếu v $\neq$ s, t.

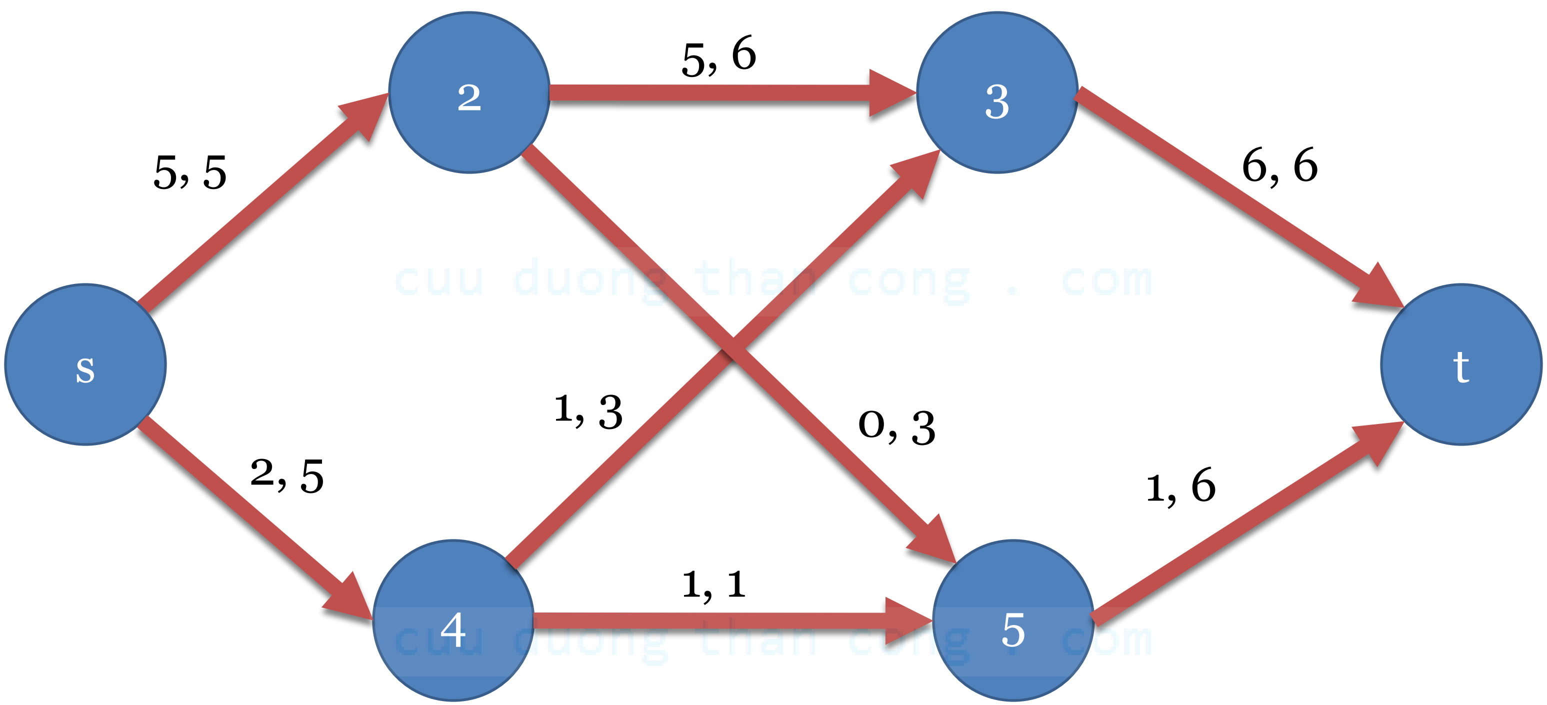
    $$t(W^-(x)) = t(W^+(x)),\ \forall x \neq s, t$$

Các định nghĩa – Giá trị của luồng

- Giá trị của một luồng được tính bằng tổng giá trị trên các cung đi ra từ đỉnh nguồn s, hoặc tổng giá trị trên các cung đi vào đỉnh thu t.

$$\text{val}(f) = t(W^{+}(s)) = t(W^{-}(t))$$

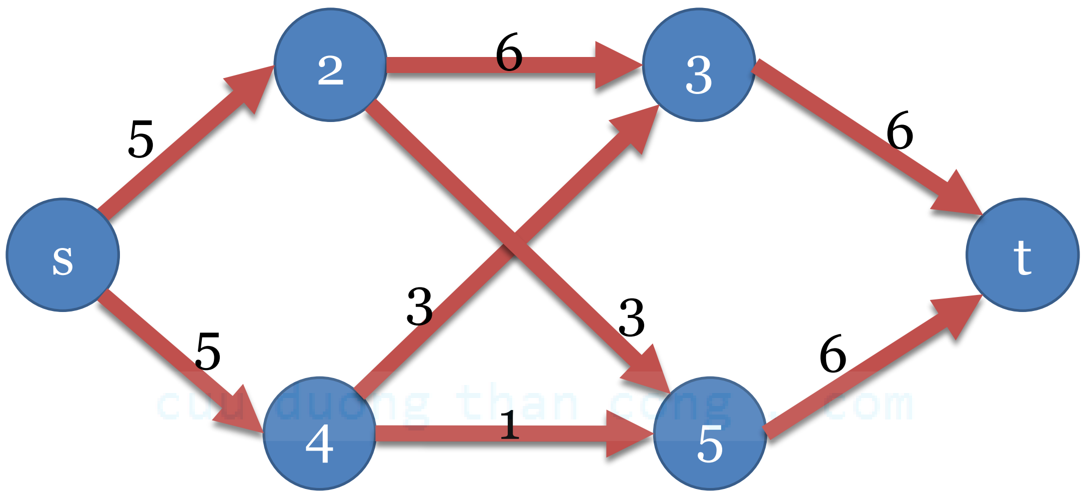
Ví dụ

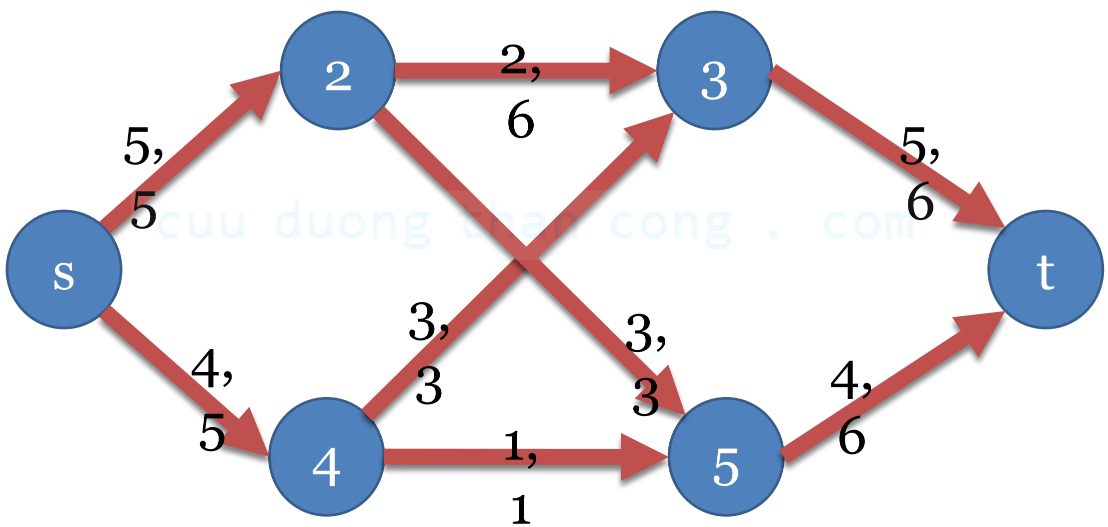


Bài toán luồng cực đại

- Cho một mạng G = (V, E), hãy tìm luồng f* trong mạng với giá trị luồng val(f*) là lớn nhất. Luồng như vậy sẽ được gọi là luồng cực đại trong mạng

Bài toán luồng cực đại – Ví dụ



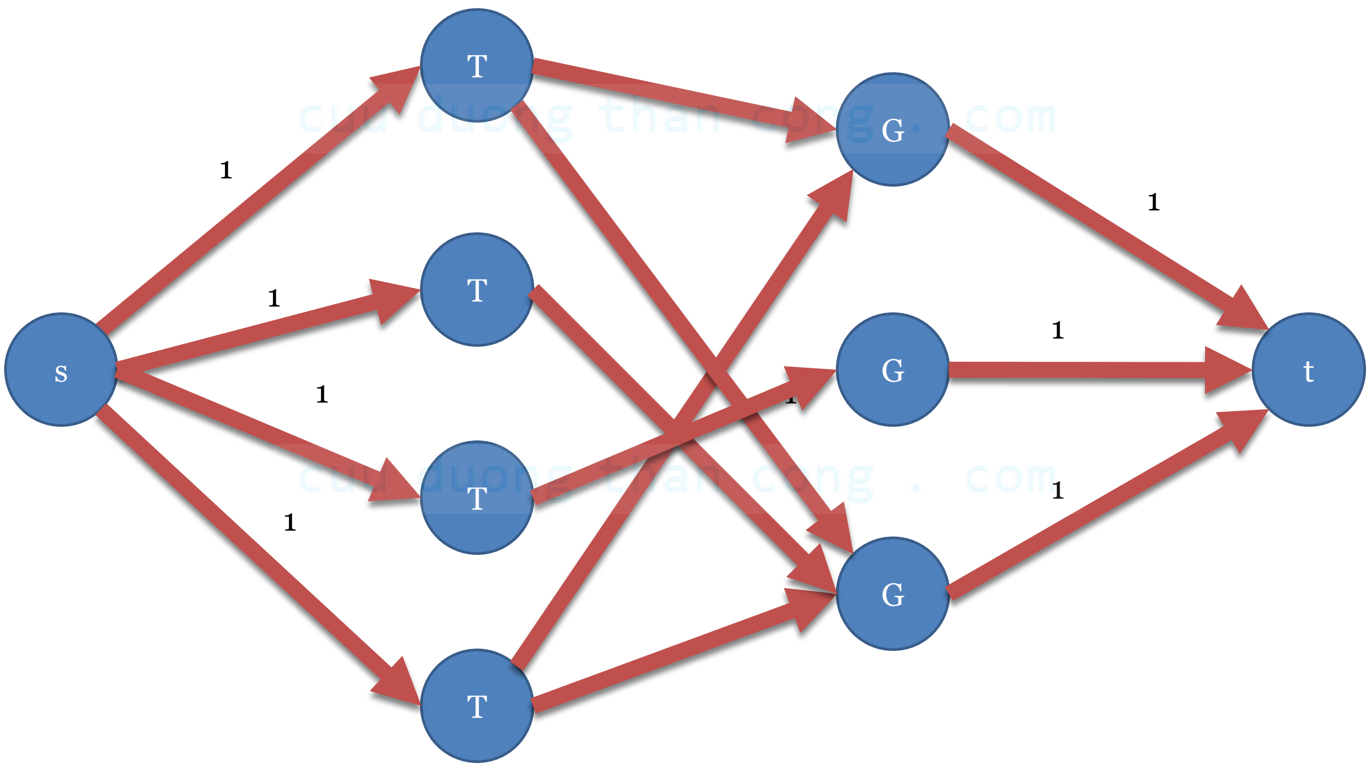


Một số ví dụ tiêu biểu

- Xét đồ thị tương ứng hệ thống ống dẫn dầu. Trong đó các ống tương ứng với các cung, điểm phát là tàu chở dầu, điểm thu là bể chứa, các điểm nối của ống là các nút của đồ thị. Khả năng thông qua của các cung tương ứng là tiết diện các ống.
- Cần phải tìm luồng dầu lớn nhất có thể bơm từ tàu chở dầu vào bể chứa.

Một số ví dụ tiêu biểu

- Bài toán cặp ghép: có m chàng trai và n cô gái. Mỗi chàng trai ưa thích một số cô gái. Hãy tìm cách ghép cặp sao cho số cặp ghép được là nhiều nhất.



Lát cắt

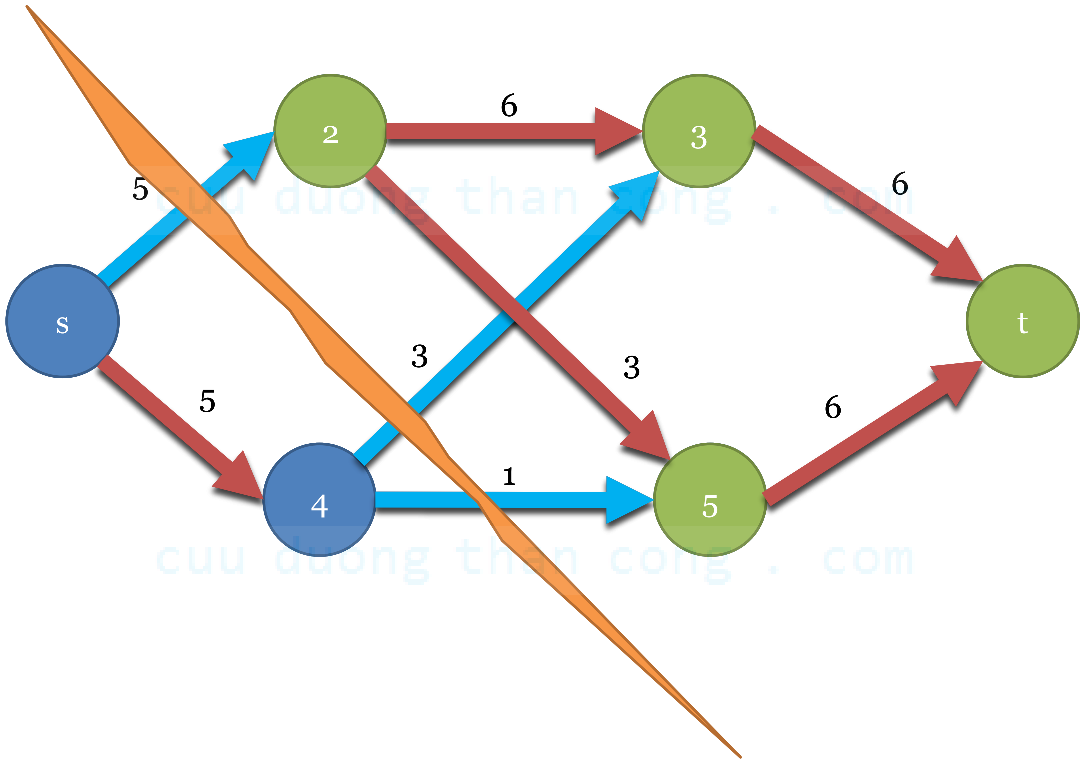
- Ta gọi lát cắt (X, X*) là một cách phân hoạch tập đỉnh V của mạng ra thành 2 tập X và X* = V\X, trong đó s $\in$ X và t $\in$ X*. Khả năng thông qua của lát cắt (X, X*) là số

$$c(X, X^*) = \sum_{\substack{v \in X \\ u \in X^*}} c(v, u)$$

- Lát cắt mà khả năng thông qua nhỏ nhất gọi là **lát cắt hẹp nhất**.

Lát cắt – ví dụ

- Xác định lát cắt hẹp nhất của mạng sau:



Max flow – min cut

- Giá trị của mọi luồng f trong mạng luôn nhỏ hơn hoặc bằng khả năng thông qua của lát cắt (X, X*) bất kỳ trong nó:

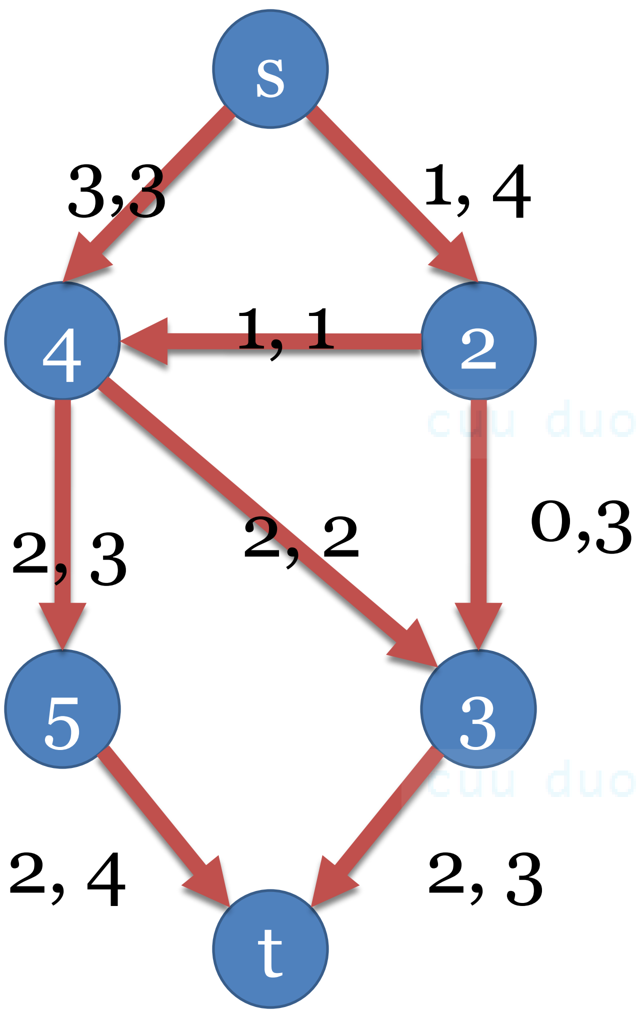
$$val(f) \leq c(X, X^*)$$

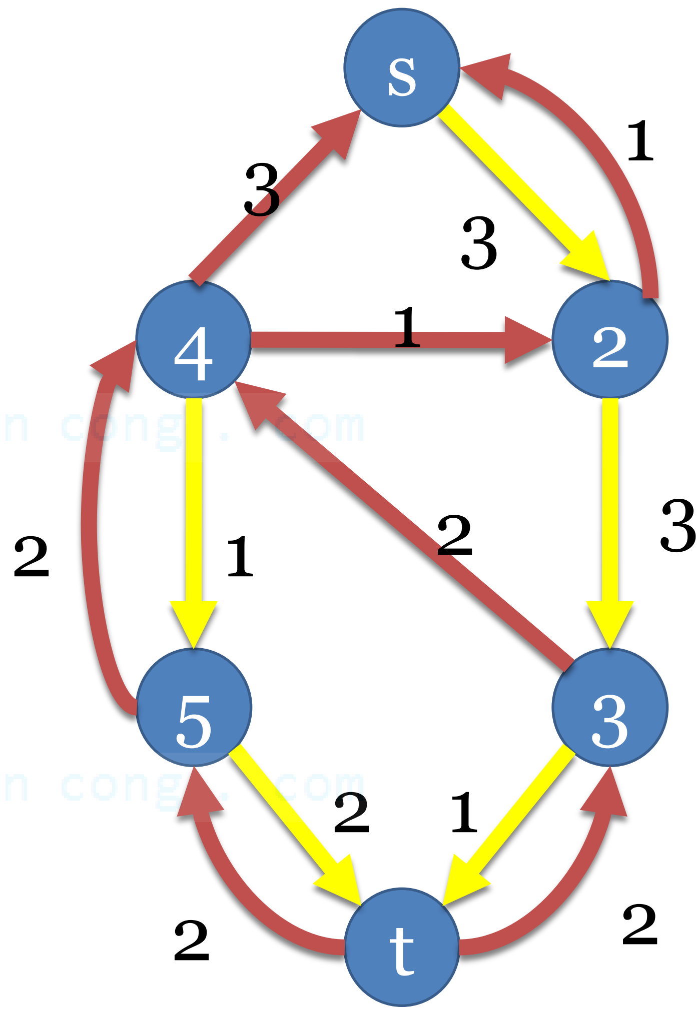
- Từ đó suy ra: Giá trị luồng cực đại trong mạng không vượt quá khả năng thông qua của lát cắt hẹp nhất trong mạng.
- Định lý: **Giá trị luồng cực đại trong mạng bằng khả năng thông qua của lát cắt hẹp nhất**.

Đồ thị tăng luồng

- Giả sử f là một luồng trên mạng G = (V, E). Từ mạng G = (V, E) ta xây dựng đồ thị có trọng số trên cung $G_f = (V, E_f)$ với tập cung E_f và trọng số trên các cung được xác định theo quy tắc sau:
  1. Nếu $e = (u, v) \in E$ với $f(u, v) = 0$ thì $(u, v) \in E_f$ với trọng số $c(u, v)$.
  2. Nếu $e = (u, v) \in E$ với $f(u, v) = c(u, v)$ thì $(v, u) \in E_f$ với trọng số $f(u, v)$.
  3. Nếu $e = (u, v) \in E$ với $0 < f(u, v) < c(u, v)$ thì $(u, v) \in E_f$ với trọng số $c(u, v) – f(u, v)$ và $(v, u) \in E_f$ với trọng số $f(u, v)$.
- Các cung của G_f đồng thời cũng là cung của G được gọi là ***cung thuận***, các cung còn lại được gọi là ***cung nghịch***. Đồ thị G_f được gọi là ***đồ thị tăng luồng***.

Đồ thị tăng luồng – Ví dụ





Tăng luồng

- Giả sử P = (s = v_0, v_1, v_2… v_k = t) là một đường đi từ s đến t trên đồ thị tăng luồng G_f. Gọi k là trọng số cung nhỏ nhất trên đường đi P. Xây dựng luồng f' theo quy tắc sau:
  - f'(u, v) = f(u, v) + k, nếu (u, v) $\in$ P là cung thuận.
  - f'(u, v) = f(u, v) – k, nếu (u, v) $\in$ P là cung nghịch.
  - f'(u, v) = f(u, v) nếu (u, v) $\notin$ P.
- Dễ dàng kiểm tra được rằng f' xây dựng như trên là luồng trong mạng và val(f') = val(f) + k.
- Thủ tục tăng luồng này gọi là **tăng luồng dọc theo đường P**.

Đường tăng luồng

- Đường tăng luồng f là mọi đường đi từ s đến t trên đồ thị tăng luồng G_f.
- Các mệnh đề dưới đây là tương đương:
  1. f là luồng cực đại trong mạng.
  2. Không tìm được đường tăng luồng f.
  3. val(f) = c(X, X*) với một lát cắt (X, X*) nào đó.

Chứng minh

- CM 2 ➔ 3: Ký hiệu X là tập các đỉnh đến được từ s, Đặt X* = V\X. Lúc đó (X, X*) là lát cắt và f(u, v) = 0 với mọi $u \in X^*$ và $v \in X$. Do đó:

$$val(f) = \sum_{\substack{u \in X \\ v \in X^*}} f(u,v) - \sum_{\substack{u \in X^* \\ v \in X}} f(u,v) = \sum_{\substack{u \in X \\ v \in X^*}} f(u,v)$$

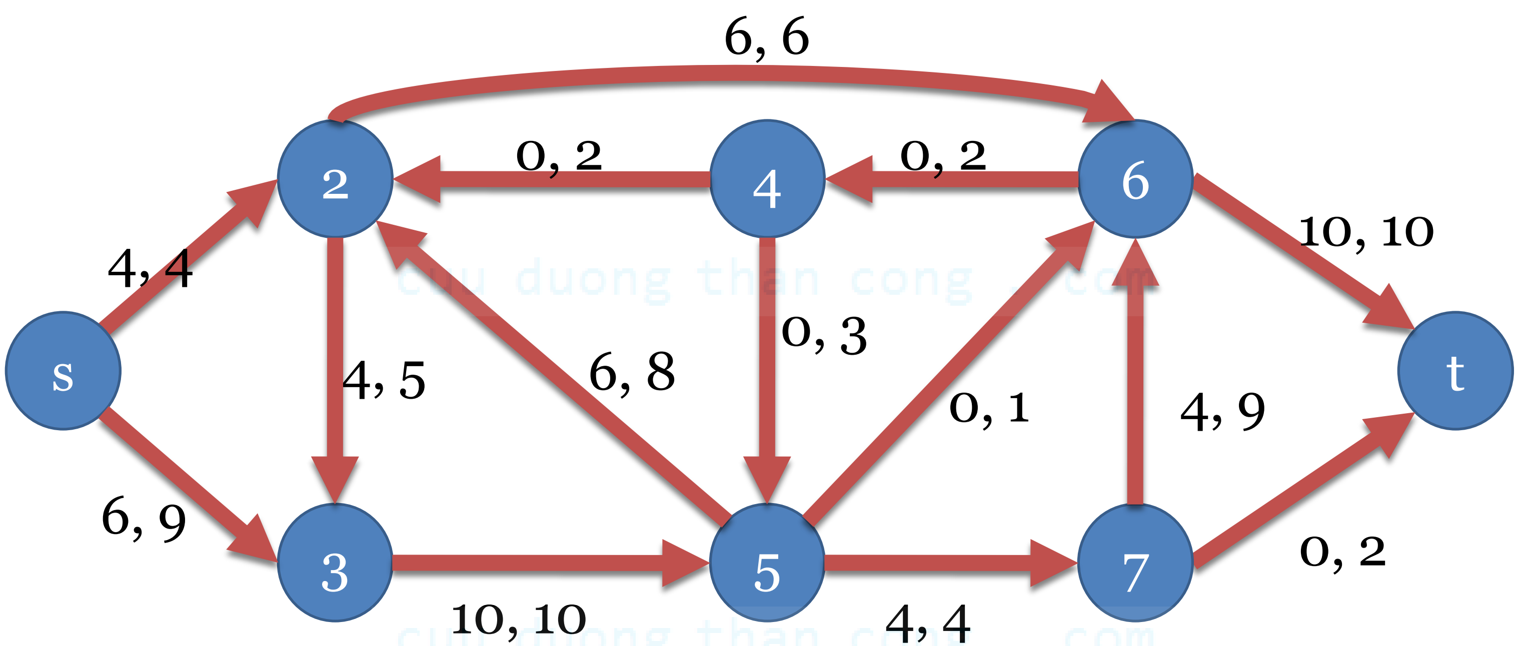
với $u \in X^*$ và $v \in X$, do $(u, v) \notin G_f$ nên f(u, v) = c(u, v). Vậy:

$$val(f) = \sum_{\substack{u \in X \\ v \in X^*}} f(u,v) = \sum_{\substack{u \in X \\ v \in X^*}} c(u,v) = c(X, X^*)$$

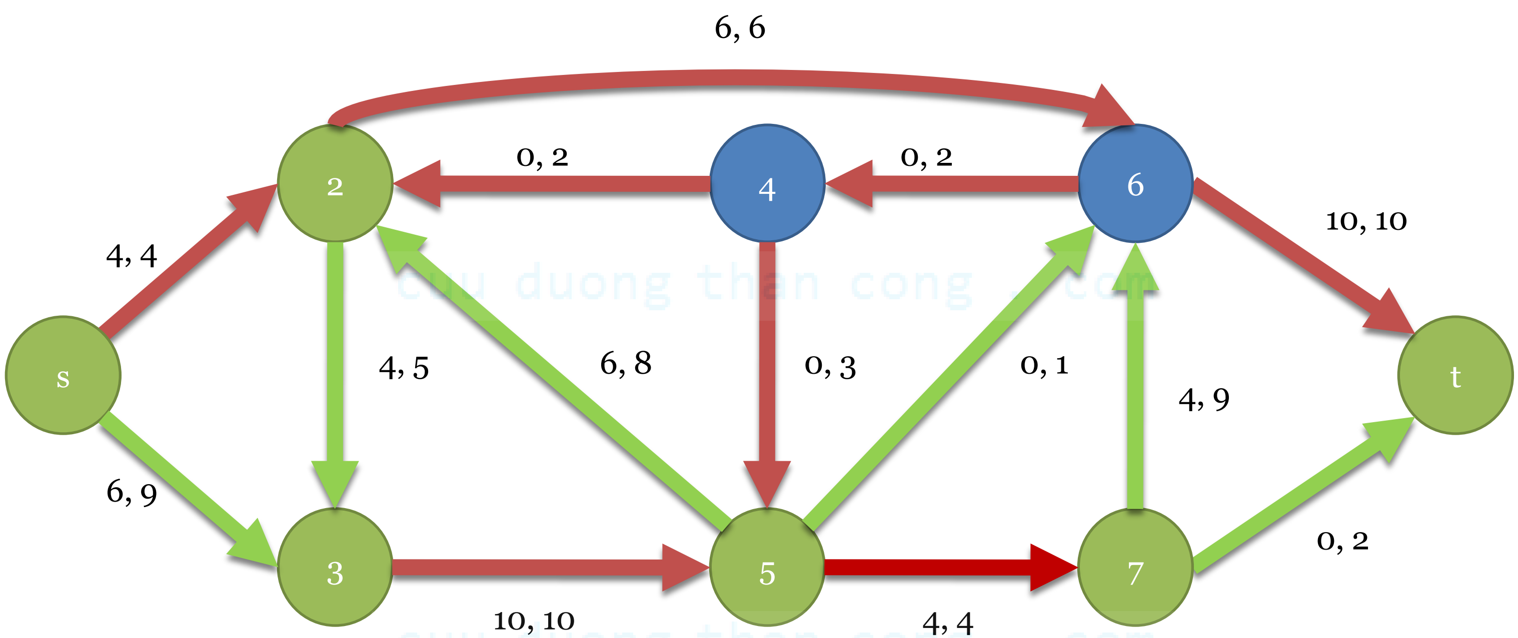
Thuật toán Ford – Fulkerson

- **Bước khởi tạo:** Bắt đầu từ luồng với luồng trên tất cả các cung bằng 0 (ta sẽ gọi luồng như vậy là luồng không), và lặp lại bước lặp sau đây cho đến khi thu được luồng mà đối với nó không còn đường tăng.
- **Bước lặp tăng luồng (Ford – Fulkerson):** Tìm đường tăng P đối với luồng hiện có. Tăng luồng dọc theo P.
- Khi đã có luồng cực đại, lát cắt hẹp nhất có thể tìm theo thủ tục mô tả trong chứng minh trước.

Ví dụ

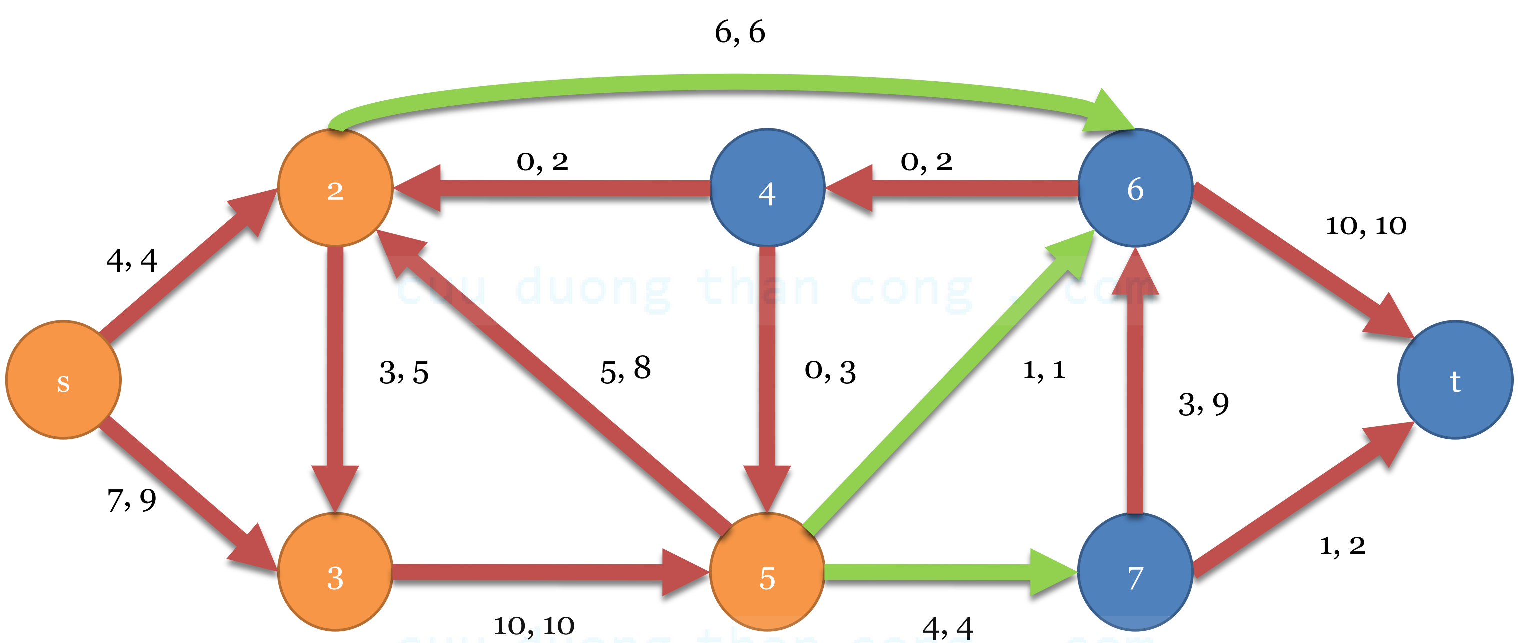


Ví dụ



$k(t) = 1$

Ví dụ



Bản chất đồ thị tăng luồng

- Bản chất của đồ thị tăng luồng là gì?

Cài đặt thuật toán Ford – Fulkerson

- Có cần xây dựng đồ thị tăng luồng?
  - Không cần xây dựng tường minh.
- Tìm đường tăng luồng theo giải thuật nào?
  - Tìm đường đi theo chiều sâu
  - Tìm đường theo chiều rộng
  - Tìm đường đi ngắn nhất

Thuật toán gán nhãn tìm đường tăng luồng

- Mỗi đỉnh sẽ có 1 trong 3 trạng thái: chưa có nhãn, có nhãn chưa xét, có nhãn đã xét.
- Nhãn của một đỉnh gồm có 2 phần:
  - p(v): Đỉnh trước của đỉnh v trên đường tăng luồng tìm được.
  - e(v): chỉ ra lượng lớn nhất có thể tăng (giảm).
- Đầu tiên, khởi tạo bằng cách gán nhãn đỉnh s là chưa xét, các đỉnh khác chưa có nhãn.
- Từ các đỉnh đã có nhãn chưa xét, gán nhãn cho tất cả các đỉnh chưa có nhãn kề với nó và nhãn của đỉnh v trở thành đã xét. (*)
- Kết thúc khi t được gán nhãn hoặc không đến được t.

Thuật toán gán nhãn tìm đường tăng luồng

- Cập nhật như thế nào ở phần (*)?
  - Nếu c[u, v] > 0 và f[u, v] < c[u, v] // còn có thể tăng!
    - p[v] = u.
    - e[v] = min{e[u], c[u, v] – f[u, v]}
  - Nếu c[v, u] > 0 và f[v, u] > 0 // thử quay ngược lại
    - p[v] = -u. // để xác định cung thuận/nghịch
    - e[v] = min{e[u], f[v, u]}

Hàm tăng luồng

```
v = p[t]; u = t; tang = e[t];
while (u ≠ s) {
    if (v > 0) {
        f[v, u] += tang;
    } else {
        v = -v;
        f[u, v] = f[u, v] - tang;
    }
    u = v;
    v = p[u];
}
```

Mã nguồn (cải tiến)

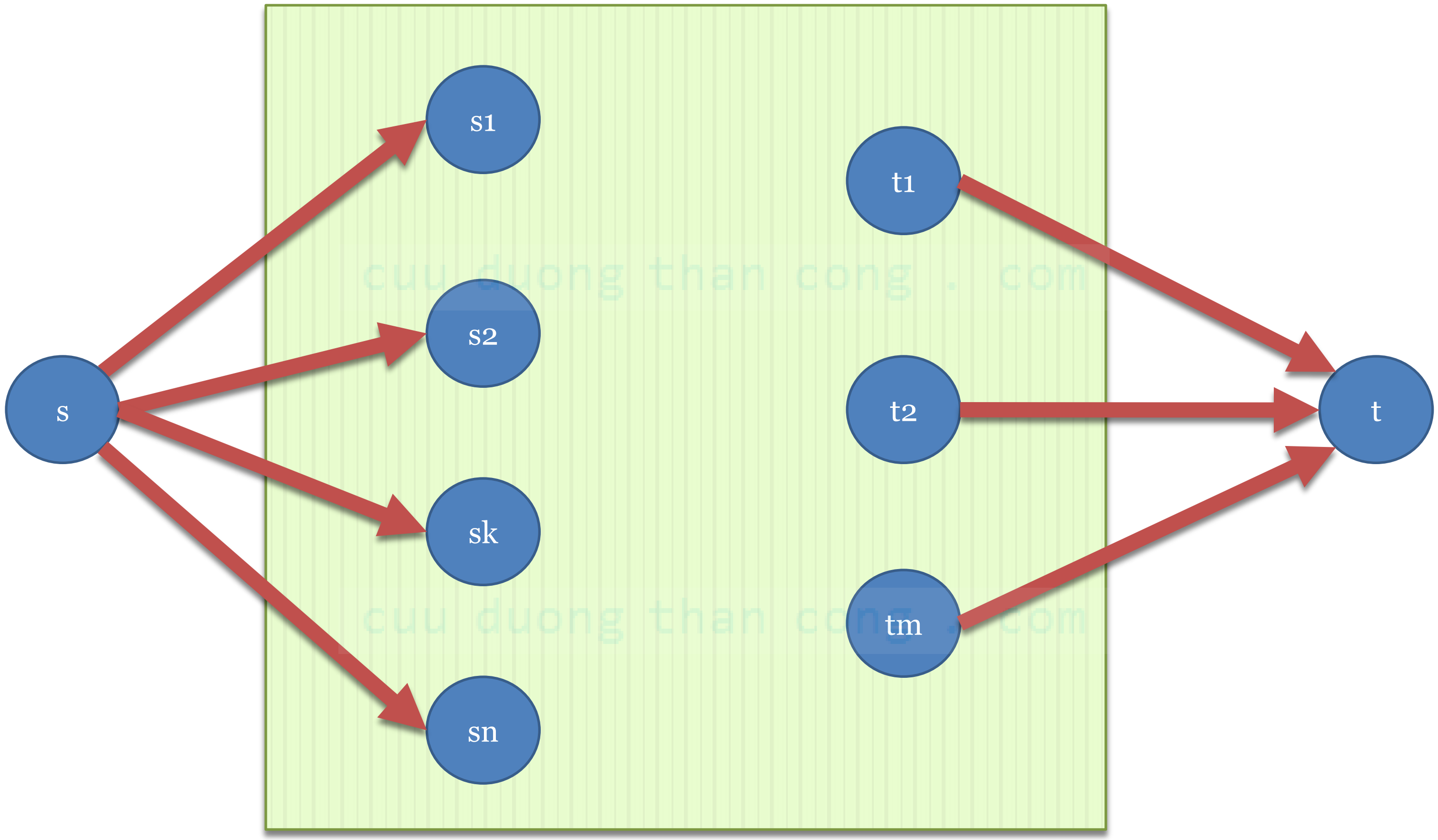
```
void Flow(int u, e)
    visit[u] = True;
    if (u == t)
        incF = e;
        return;
    if (visit[t]) return;
    for (int i = 0; i < n; i++)
        if (!Visit[i])
            if (c[u, i] > f[u, i])
                Flow(i, min(e, c[u, i] - F[u, i]));
                if (visit[t])
                    F[u, i] += incF;
                    return;
            if (F[i, u] > 0)
                Flow(i, min(e, F[i, u]));
                if (visit[t])
                    F[i, u] -= incF;
                    return;
```

Mã nguồn (cải tiến)

```
void MaxFlow()
{
      FillZero(F, sizeof(F), 0);
      do {
            FillZero(visit, sizeof(visit), 0);
            Flow(s, +∞);
      while (!visit[t]);
}
```

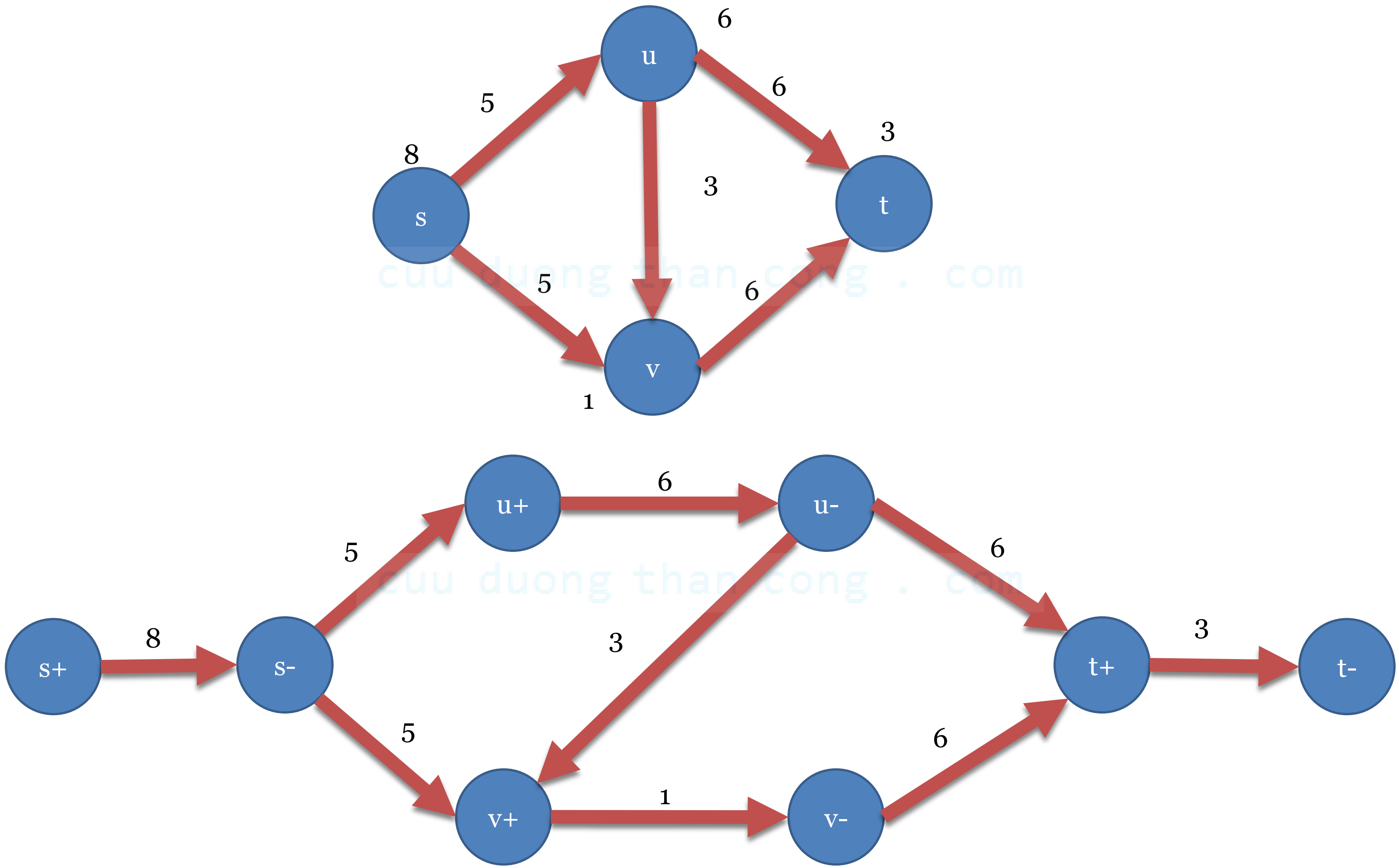
Các dạng bài toán luồng

- Mạng với nhiều điểm phát và điểm thu



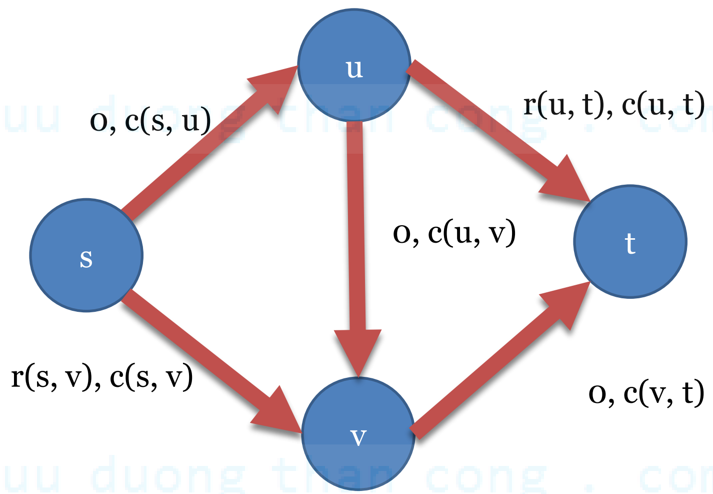
Các dạng bài toán luồng

- Khả năng thông qua của cung và đỉnh



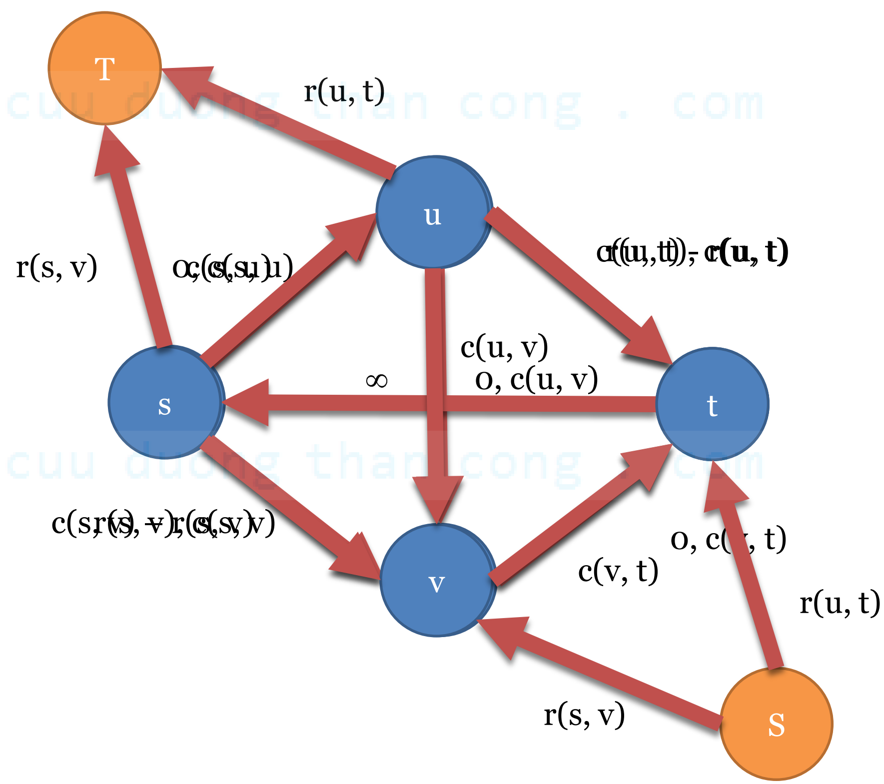
Các dạng bài toán luồng

- Khả năng của cung bị chặn 2 phía



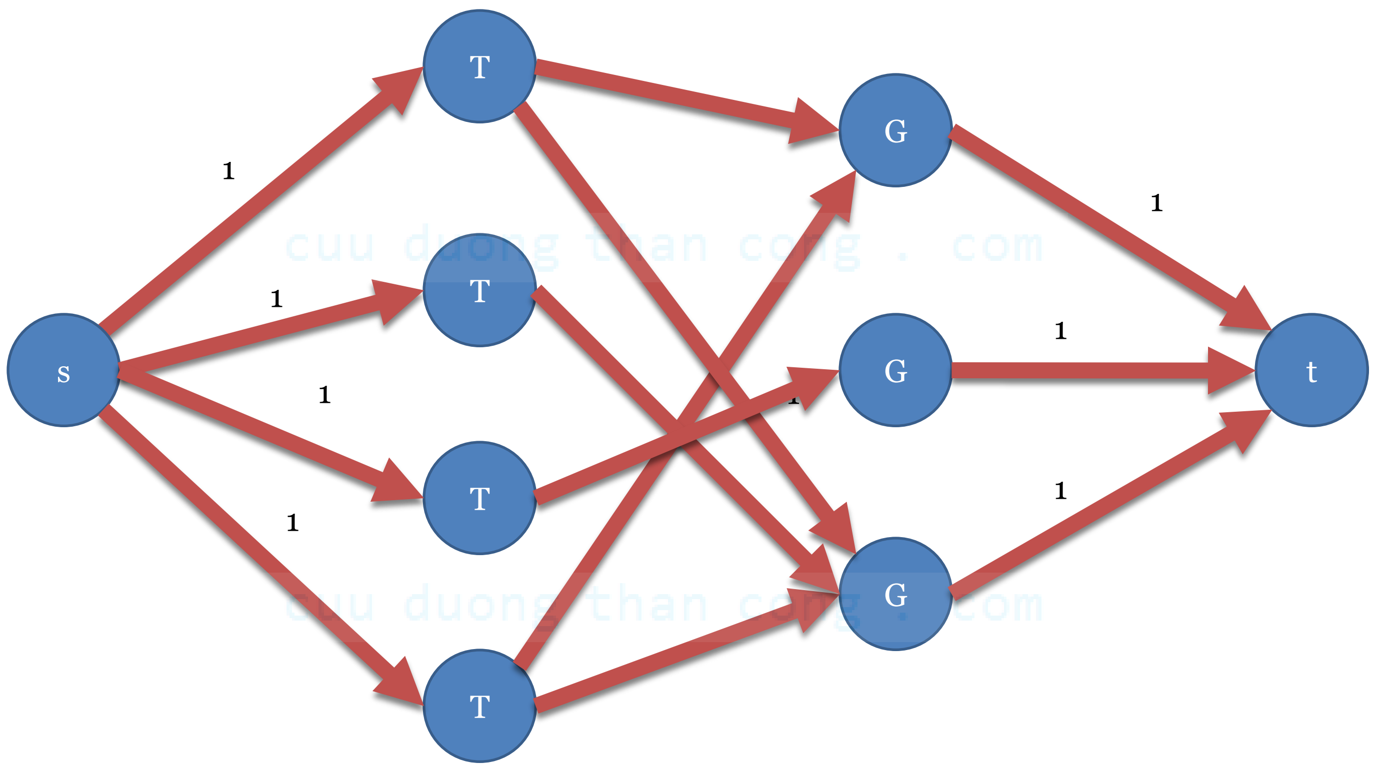
Các dạng bài toán luồng

- Cung r(u, v) ≠ 0, thay bằng (**S**, v) và (u, **T**) = r(u, v).
- c(u, v) = c(u, v) – r(u, v)
- Thêm cung (t, s) với c(t, s) = ∞



Một số ứng dụng

- **Bài toán đám cưới vùng quê (bài toán cặp ghép)**



Một số ứng dụng

- **Bài toán hệ thống đại diện chung:** Cho tập gồm m phần tử $X = \{z_1, z_2 \ldots z_m\}$. Giả sử $<A_1, A_2 .. A_n>$ và $<B_1, B_2 \ldots B_n>$ là 2 dãy các tập hợp con của X.
- Dãy gồm n phần tử $x_1, x_2 \ldots x_n$ được gọi là hệ thống đại diện chung của 2 dãy A và B đã cho nếu như tìm được một hoán vị k của tập $\{1, 2, \ldots n\}$ sao cho $<x_1, x_2, \ldots x_n>$ là các đại diện phân biệt của 2 dãy $<A_1, A_2, \ldots A_n>$ và $<B_{k(1)}, B_{k(2)}, \ldots B_{k(n)}>$ nêu trên.

Một số ứng dụng

- **Bài toán hệ thống đại diện chung – Ví dụ**
- Một giải vô địch bóng đá có M cầu thủ đăng ký thi đấu. Mỗi cầu thủ của giải bắt buộc phải thuộc ít nhất một câu lạc bộ bóng đá và một hội lao động.
- Có N câu lạc bộ A_1, A_2.. A_n và N hội B_1, B_2… B_n.
- Ban tổ chức giải và liên đoàn lao động cần chọn ra một ban đại diện gồm N cầu thủ sao cho mỗi câu lạc bộ và mỗi hội lao động đều có ít nhất một thành viên của mình trong ban đại diện này.
- Hãy tìm một ban đại diện như vậy.

Một số ứng dụng

- **Các bài toán tối ưu rời rạc – Bài toán luồng chi phí nhỏ nhất**
- Cho một mạng có n đỉnh. Mỗi cạnh của mạng có một khả năng thông qua c(u, v) và một cước phí vận chuyển p(u, v) nhất định ứng với một đơn vị hàng.
- Cho trước một lượng hàng S cần vận chuyển từ đỉnh nguồn đến đỉnh đích, tìm phương án tối ưu để chi phí vận chuyển hết lượng hàng S là nhỏ nhất.

Một số ứng dụng

- Các bài toán tối ưu rời rạc – Luồng chi phí nhỏ nhất
  - Nếu không cần vận chuyển cùng lúc?
  - Nếu cần vận chuyển cùng lúc?

Một số ứng dụng

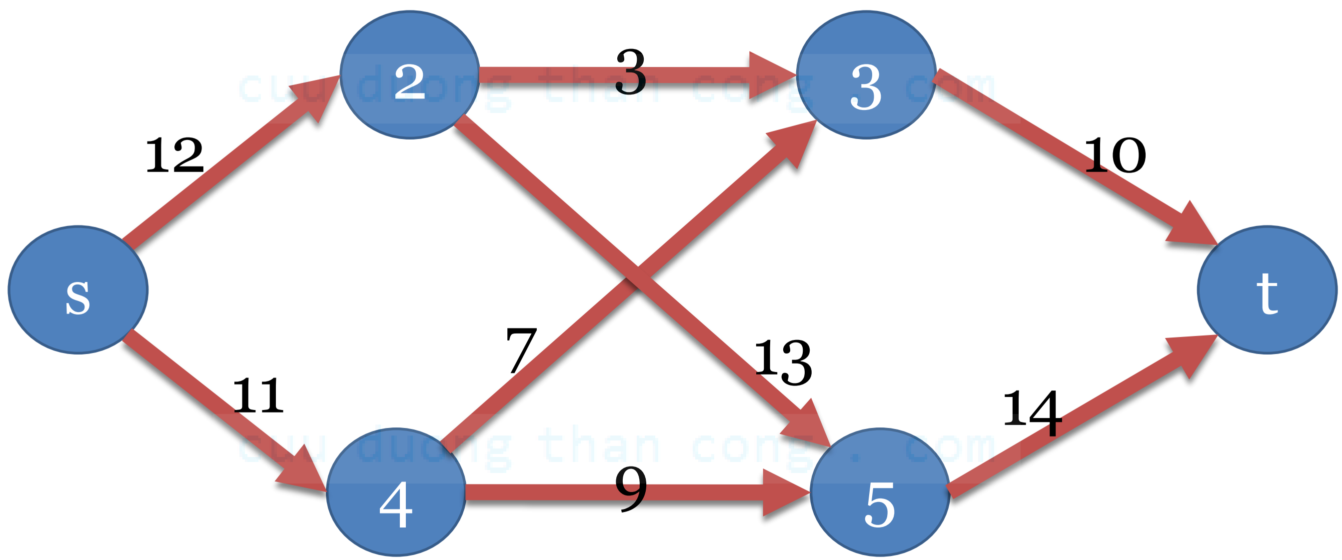
- **Khả năng liên thông cạnh (Edge Connectivity)** của một đồ thị vô hướng là số cạnh cực tiểu k, sao cho khi gỡ bỏ đi k cạnh này thì đồ thị sẽ không còn liên thông nữa.
- Ví dụ: Đối với đồ thị dạng cây, k = 1. Với đồ thị là chu trình, k = 1.

Một số ứng dụng

- **Đường đi bao phủ** (path cover) của một đồ thị vô hướng G = (V, E) là một tập hợp P các đường đi rời nhau trên G sao cho mọi đỉnh trong V nằm chính xác trong một đường đi của P.
- Các đường đi có thể bắt đầu và kết thúc ở bất kỳ đâu, và có thể có bất kỳ chiều dài nào (nghĩa là có thể có 1 đỉnh duy nhất hoặc toàn bộ đỉnh của V).
- Một *path cover cực tiểu* của G là một path cover có ít đường đi nhất.

Bài tập

1. Tìm luồng cực đại cho mạng sau:



Bài tập

2. Hãy nêu giải phát để giải quyết vấn đề liên thông cạnh.
3. *Hãy nêu giải pháp tìm được path cover cực tiểu.
4. Chứng minh rằng một luồng cực đại trên mạng G = (V, E) luôn có thể xác định được sau một dãy tối đa |E| quá trình tìm đường tăng luồng.
5. Chứng minh với một cặp đỉnh u, v bất kỳ, ta luôn có $c_f(u, v) + c_f(v, u) = c(u, v) + c(v, u)$. Với c_f là trọng số của cung trên đồ thị tăng luồng.