Đồ thị

Phan Minh Hoàng

June 2017

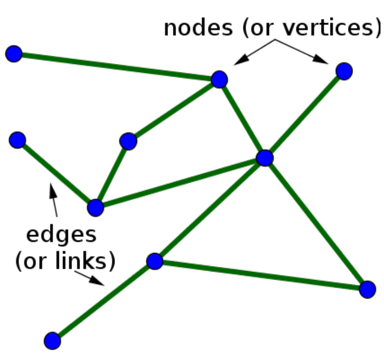
Tóm tắt nội dung

Trong cuộc sống, chúng ta rất hay gặp những bài toán liên quan đến các đồ vật và việc biểu thị quan hệ giữa những đồ vật khác nhau. Vì vậy, các nhà toán học đã đưa ra một mô hình toán học chung để giải quyết những bài toán như vậy đó chính là đồ thị. Do sự gắn bó chặt chẽ của nó với các bài toán thực tế nên đây là một phần được nghiên cứu rất kĩ trong khoa học máy tình và toán rời rạc.

1 Đồ thị là gì ?

1.1 Đỉnh và cạnh

Để biểu diễn một đồ thị G và các mối quan hệ ta dùng hai tập một là tập các đồ vật mà ta cần biểu diễn quan hệ (học sinh, điểm giao trên đường) và quan hệ của chúng (bạn bè, các đường nối). Các đồ vật được gọi là tập các **đỉnh** V và quan hệ là tập các **cạnh** E. Nếu e là một cạnh của đồ thị thì e sẽ có ít nhất hai tham số u và v biểu diễn hai đỉnh mà cạnh nối. Tuỳ vào mô hình các cạnh này có thể có thêm những tham số khác như **hướng** hay **trọng số**. Trong đồ thị **vô hướng** nếu tồn tại một cạnh e nối hai đỉnh u và v ta nói u và v **kề nhau** và u và v **liên thuộc** e.



Hình 1: Đỉnh và cạnh.

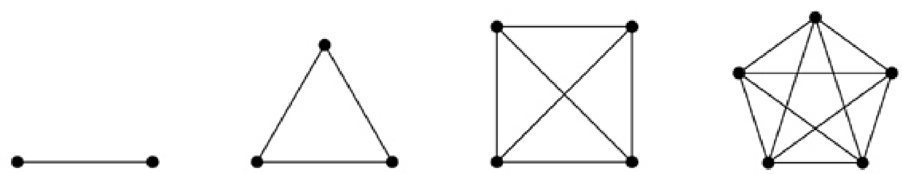
1.2 Bậc

Bậc của đỉnh v hay deg_v là số cạnh liên thuộc với v. Tổng các bậc này được gọi là deg_G hay bậc của đồ thị G và bằng $2 * M$ với M là số cạnh. Bậc của một đỉnh bị **cô lập** là 0. Số đỉnh có bậc lẻ luôn là chẵn.

1.3 Các loại đồ thị đặc biệt

1.3.1 Đồ thị đầy đủ

Đây là loại đồ thị mà tất cả các đỉnh đều có quan hệ với nhau và theo tổ hợp cơ bản số cạnh của một đồ thị đầy đủ N đỉnh là $\binom{N}{2}$.

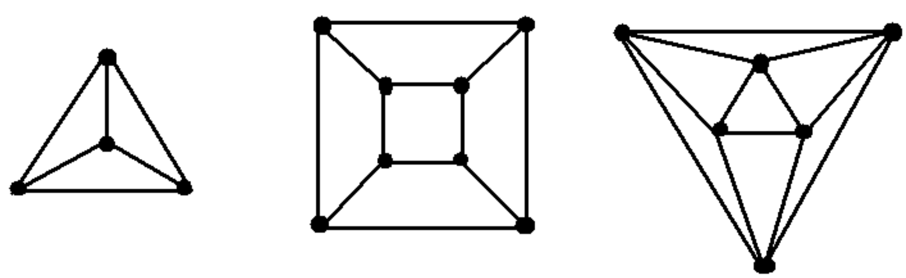


Hình 2: Đồ thị đầy đủ

1.3.2 Đồ thị phẳng

Đây có lẽ là loại đồ thị có ứng dụng cao trong đời sống do tính chất đặc biệt của nó là có thể biểu diễn được đồ thị này trên mặt phẳng mà không có hai cạnh nào cách cắt nhau ngoài trừ trên cạnh.

Khi biểu diễn đồ thị này trên mặt phẳng ta có thể tính được số miền mà các cạnh này chia ra theo công thức $F = M - N + 1$ với M là số cạnh, N là số đỉnh và F là số miền (Tự chứng minh). Việc này rất hữu dụng trong việc đếm sồ miền được chia bởi các đường thẳng.

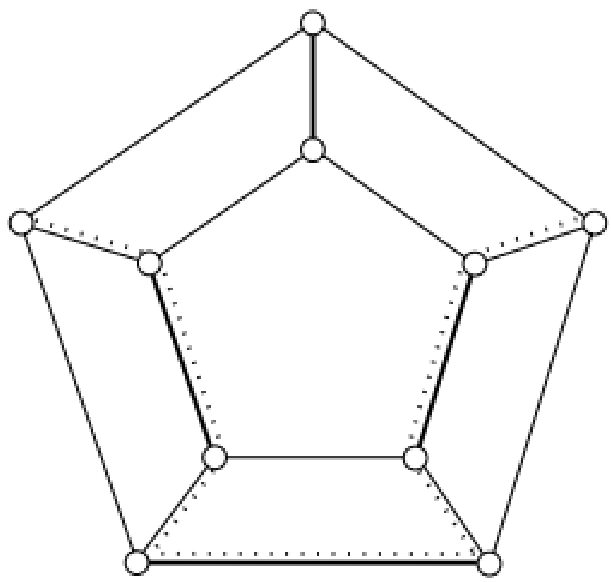


Hình 3: Đồ thị phẳng

2 Một số khái niệm quan trọng

2.1 Đường đi và chu trình

Một đường đi từ u đến v là một dãy các đỉnh $v_0, v_1, v_2, ...v_k$ trong v_i có cạnh nối trực tiếp v_{i+1} với $i < k$, $v_0 = u$ và $v_k = v$. Một chu trình là một đường đi với $v_0 = v_k$. Trong một số trường hợp chu trình hoặc đường đi có thể đòi hỏi không đi trùng cạnh hoặc đỉnh.

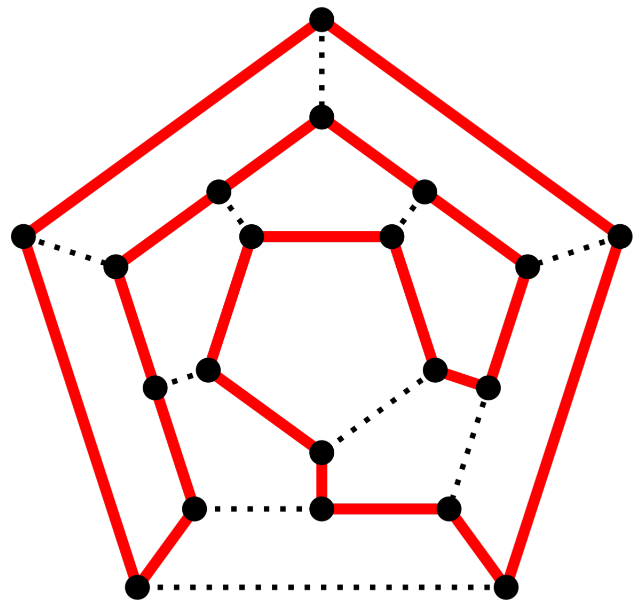


Hình 4: Đường đi trên đồ thị.

2.1.1 Đường đi và chu trinh Hamilton

Một chu trình Hamilton là một chu trình đi qua tất cả các đỉnh mỗi đỉnh một lần và quay lại đỉnh xuất phát. Đường đi Hamilton là một đường đi qua tất cả các đỉnh mỗi đỉnh một lần. Bài toán kiểm tra đồ thị có tồn tại chu trình hoặc đường đi Hamilton là một bài toán thuộc lớp NP đầy đủ, tức chưa có thuật toán đa thức để giải quyết bài toán này.

Tuy vậy định lý Dirac đã chỉ ra rằng đồ thị có bậc nhỏ nhất lớn hơn hoặc bằng một nửa số đỉnh ($min\ deg_v \geq N/2$ với $v \in V$) luôn tồn tại chu trình Hamilton.

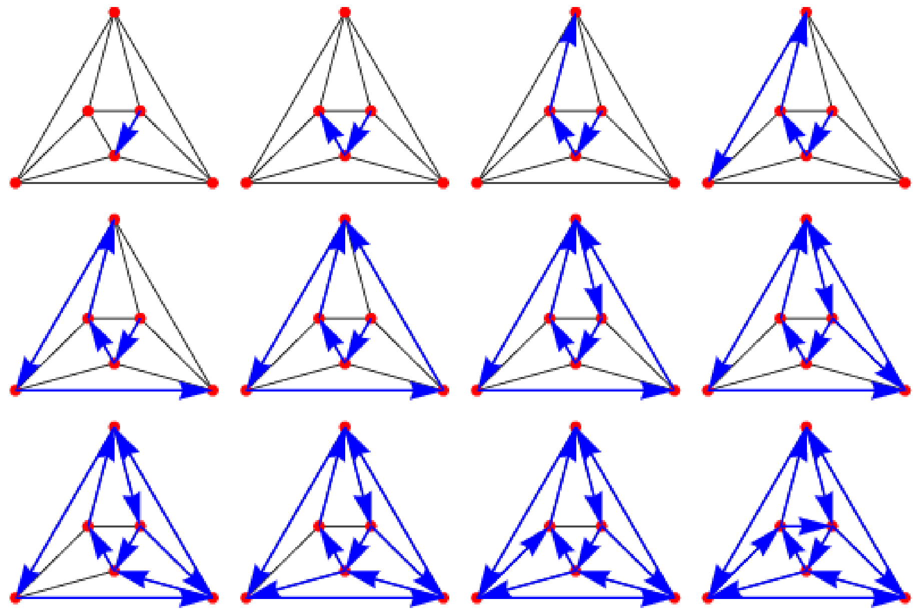


Hình 5: Đường đi Hamilton.

2.1.2 Chu trình Euler

Một chu trình Euler là một chu trình đi qua tất cả các cạnh mỗi cạnh một lần và quay lại đỉnh xuất phát. Đường đi Euler là một đường đi qua tất cả các cạnh mỗi đỉnh một lần.

Không như chu trình Hamilton, tồn tại một thuật toán đa thức để tìm kiếm và kiểm tra tồn tại một chu trình hoặc đường đi Euler. Gọi V' là tập đỉnh trong đồ thị liên thuộc với ít nhất một cạnh. Điều kiện đầu tiên, V' phải là một thành phần liên thông. Điều kiện để tồn tại một chu trình Euler là $deg_v mod 2 = 0$ với mọi $v \in V'$ (a mod b là phép lấy phần dư khi chia a cho b). Điều kiện để tồn tại một đường đi Euler là tồn tại nhiều nhất hai đỉnh $v \in V'$ sao cho $deg_v mod 2 = 1$ với mọi $v \in V'$.



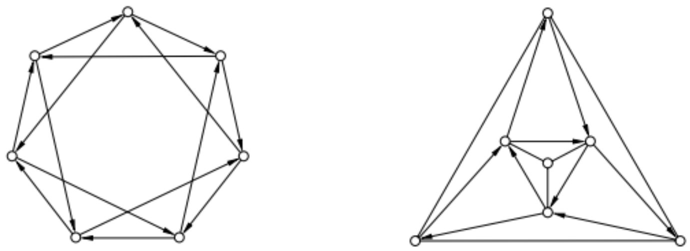
Hình 6: Chu trình Euler.

2.2 Thành phần liên thông

Trong đồ thị vô hướng, một thành phần liên thông là một tập tối đa các đỉnh mà với 2 cặp đỉnh bất kì đều tồn tại đường đi đến nhau. Tồn tại một và chỉ một cách phân hoạch các đỉnh của một đồ thị thành các thành phần liên thông.

3 Đồ thị có hướng

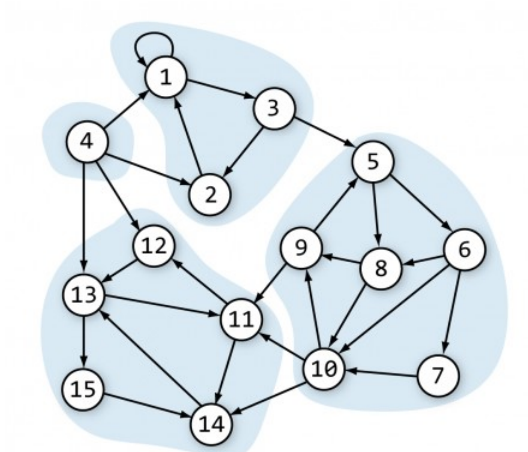
Đây là đồ thị mà mỗi cạnh chỉ có thêm một tham số là hướng tức cạnh $u \to v$ khác cạnh $v \to u$. Đồ thị bình thường là một dạng đặc biệt của đồ thị có hướng khi với mỗi cạnh từ $u \to v$ tồn tại cạnh $v \to u$ tương ứng.



Hình 7: Đồ thị có hướng.

3.1 Thành phần liên thông mạnh

Trong đồ thị có hướng nếu ta kết nối các cạnh mà không tính đến hướng đi của chúng, ta sẽ có một đồ thị vô hướng tương ứng gọi là**đồ thị khung** của đồ thị có hướng. Một đồ thị có hướng có đồ thị khung liên thông chưa hẳn là một đồ thị liên thông. Vì vậy ta có định nghĩa **liên thông mạnh** trong đồ thị có hướng là một tập các đỉnh sao cho giữa hai đỉnh bất kì tồn tại đường đi đến nhau.



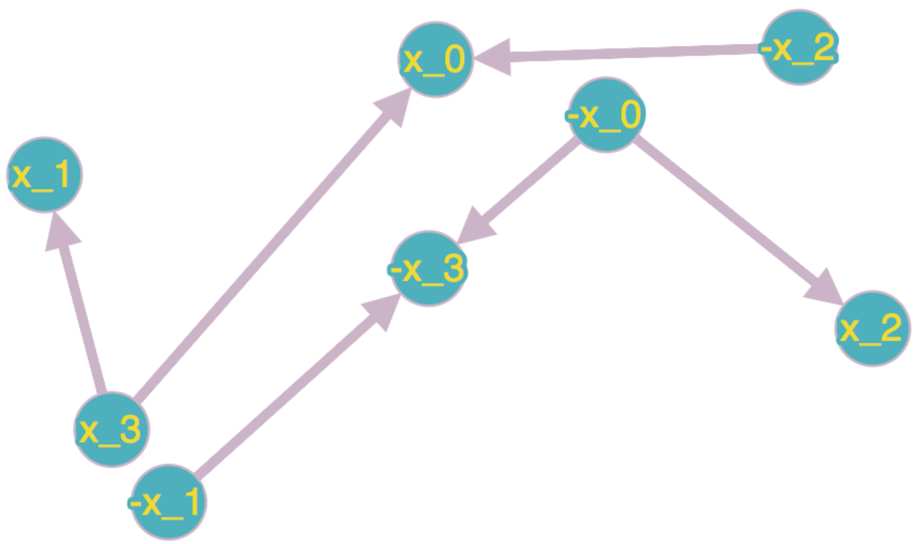
Hình 8: Thành phần liên thông.

3.2 Bài toán 2-SAT

Một ví dụ điển hình của ứng dụng đồ thị có hướng là bài toán 2-SAT hay 2-satisfiability. Cho n biến đúng sai lần lượt là x_1, x_2, ... x_n ta cần kiểm tra xem có tồn tại một cách đặt các x_i đúng hoặc sai sao cho thoả một dãy điều kiện $\wedge(\pm x_i \vee \pm\, x_j)$. Một ví dụ của điều kiện có 5 biến là $(x_1 \vee x_2) \wedge (-x_2 \vee x_3) \wedge (-x_4 \vee -x_5)$.

Ta có thể thấy một điều kiện $x_i \vee x_j$ tương đương với điều kiện sau $(-x_i \rightarrow x_j)$ và $(-x_j \rightarrow x_i)$. Ta có thể thấy đây là một điều kiện kéo theo. Vì vậy ta có thể lập một đồ thị có $2 * n$ đỉnh tương ứng với $2 * n$ điều kiện $\pm x_i$ và các cạnh nối có hướng với các điều kiện kéo theo. Do hiển nhiên điều kiện x_i và $-x_i$ không thể cùng xuất hiện, ta có thể đưa đến một cách kiểm tra cho bài toán 2-SAT có nghiệm hay không bằng cách kiểm tra xem có tồn tại một thành phần liên thông mạnh chứa cả x_i lẫn $-x_i$ hay không.

Ví dụ: Ta xét một điều kiện sau $(x_0 \vee x_2) \wedge (x_0 \vee -x_3) \wedge (x_1 \vee -x_3)$ và đồ thị 2-SAT của nó dưới đây:



Hình 9: Đồ thị 2-SAT của bài toán.

Ta có thể thấy đồ thị này là một đồ thị có hướng không chu trình nên ta có thể nói rằng tồn tại nghiệm để đồ thị này thoả. Một nghiệm có thể là $x_0 = True$, $x_1 = True$, $x_2 = True$, $x_3 = False$.

4 Đồ thị có trọng số

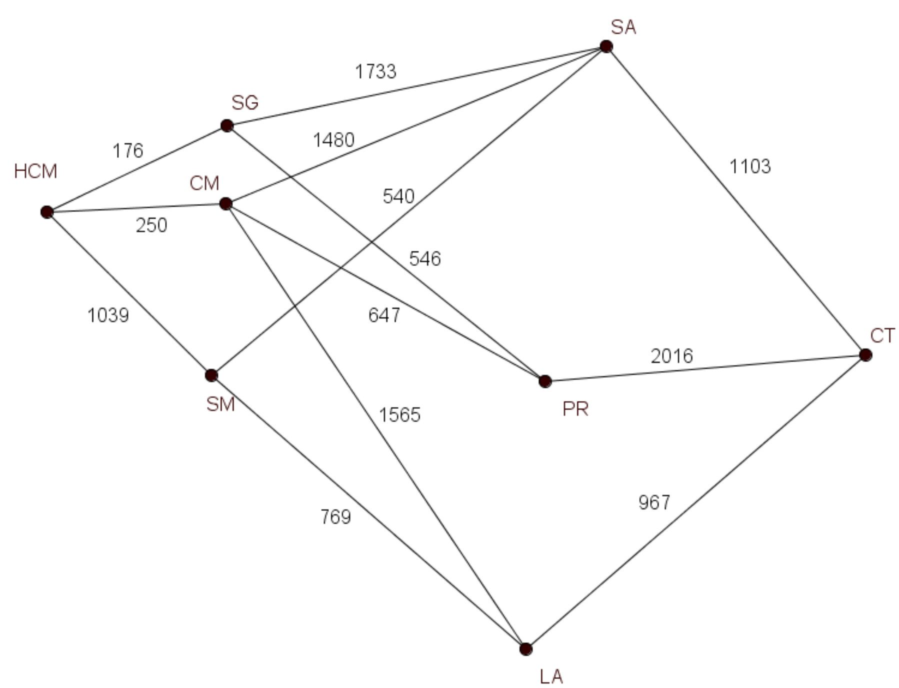
4.1 Đường đi ngắn nhất

Thầy Dũng muốn tham dự một Hội nghị về Toán mô hình ở Cape Town, Nam Phi. Tuy nhiên, do không có đường bay thẳng từ Thành phố Hồ Chí Minh đến

Nam Phi nên thầy Dũng phải quá cảnh ở hai thành phố khác. Dựa vào bảng sau đây, hãy giúp thầy Dũng chọn lộ trình bay ít tốn kém nhất.

	HCM	Chiangmai	Singapore	Santa Marta	San Antonio	Los Angeles	Paris	Cape Town
HCM	-	250	176	1039	-	-	-	-
Chiangmai	-	-	-	-	1480	1565	647	-
Singapore	-	-	-	-	1733	-	546	-
Santa Marta	-	-	-	-	540	769	-	-
San Antonio	-	-	-	-	-	-	-	1103
Los Angeles	-	-	-	-	-	-	-	967
Paris	-	-	-	-	-	-	-	2016
Cape Town	-	-	-	-	-	-	-	-

Bài toán trên có thể được đưa về mô hình đồ thị có trọng số một cách rất tự nhiên: dùng các đỉnh để biểu diễn các thành phố và các cạnh có trọng số sẽ tương ứng với giá tiền của các chuyến bay. Như vậy, về cơ bản ta cần tìm đường đi ngắn nhất từ đỉnh HCM đến đỉnh CT của đồ thị sau. ("ngắn nhất" ở đây được hiểu là đường đi có tổng trọng số nhỏ nhất, tương ứng với lộ trình bay ít tốn kém nhất)



Thuật toán Dijkstra này có nhiều nét tương đồng với thuật toán Jarnik-Prim, đặc biệt là việc áp dụng tư tưởng của thuật toán tham lam. Ta cũng chọn một đỉnh làm gốc rồi sử dụng phép tô màu để đánh dấu các đỉnh đã được chọn.

THUẬT TOÁN DIJKSTRA
input: một đồ thị liên thông có trọng số dương và một đỉnh làm gốc r.
output: hàm predecessor p và hảm khoảng cách ℓ thoả mãn $\ell(v)$ chính là độ dài của đường đi ngắn nhất từ r đến v.

1. cho $p(v) := \emptyset$ và $\ell(r) := 0, \ell(v) := \infty, v \in V \setminus \{r\}$
2. **while** có một đỉnh u chưa được tô màu mà $\ell(u) < \infty$ **do**
3. chọn đỉnh u sao cho $\ell(u)$ nhỏ nhất
4. tô màu u
5. **for** mỗi đỉnh v kề với u chưa được tô màu mà $\ell(v) > \ell(u) + w(uv)$ **do**
6. $p(v) := u$ và $\ell(v) := \ell(u) + w(uv)$
7. **end for**
8. **end while**
9. return(p, ℓ)

Hãy cùng áp dụng thuật toán Dijkstra để giúp thầy Dũng chọn đường bay tốt nhất! Ở đây đương nhiên ta sẽ chọn HCM làm gốc. Bước 2, hiển nhiên chỉ có HCM thoả mãn $\ell(HCM) < \infty$ nên ta chọn và tô màu HCM (xong bước 4). Do $\ell(CM) = \ell(SG) = \infty$ nên ở bước 5 và 6 ta lần lượt gán $p(CM) = p(SG) = HCM$ và $\ell(CM) = w(CM - HCM) = 250, \ell(SG) = w(SG - HCM) = 176$. Đến đây quay lại bước 2.
Để ý rằng chỉ có Chiangmai và Singapore là thoả mãn $\ell < \infty$, và do $\ell(SG) < \ell(CM)$ nên ta chọn Singapore tiếp theo. Xét các đỉnh kề với Singapore, hàm predecessor và hàm khoảng cách trở thành:

$$p(SA) = p(PR) = SG$$

$$\ell(SA) = 176 + 1733 = 1909,\ \ell(PR) = 176 + 546 = 722$$

Lúc này thì có ba đỉnh thoả mãn $\ell < \infty$ là Chiangmai, San Antonio Paris. Dựa vào các giá trị, dễ dàng chọn Chiangmai tiếp theo và cập nhật kết quả:

$$p(LA) = CM$$

Bây giờ chú ý là $\ell(CM) + w(CM - SA) = 1730 < \ell(SA)$ (bay Chiangmai-San Antonio thì rẻ hơn bay Singapore-San Antonio) cho nên ta thay đổi $p(SA) = CM$. Do đó:

$$\ell(LA) = 1565 + 250 = 1815,\ \ell(SA) = 1730$$

Tiếp tục như vậy, $\ell(PR)$ nhỏ nhất trong các giá trị ℓ hữu hạn đang có, cho nên đỉnh tiếp theo được chọn là Paris. p và ℓ được cập nhật:

$$p(CT) = PR,$$

$$\ell(CT) = \ell(PR) + w(PR - CT) = 722 + 2016 = 2738$$

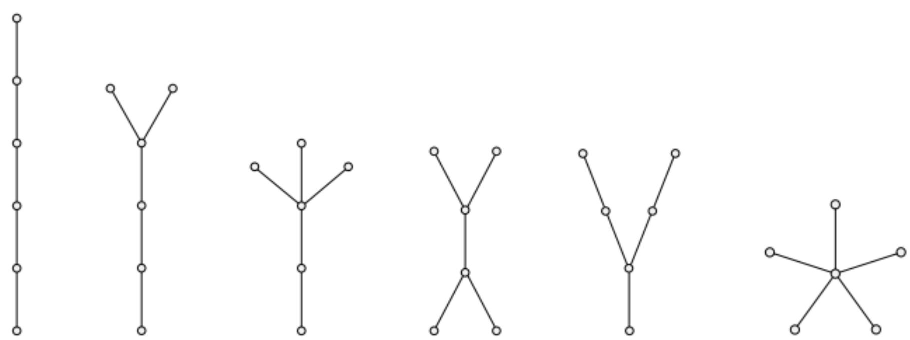
Lặp lại các thao tác trên một vài lần nữa, cuối cùng hàm p và ℓ của ta sẽ trở thành:

$$p(CM) = p(SG) = p(SM) = HCM,\ p(SA) = SM,$$

Vậy chuyến bay lí tưởng nhất là HCM-Santa Marta-San Antonio-Cape Town.

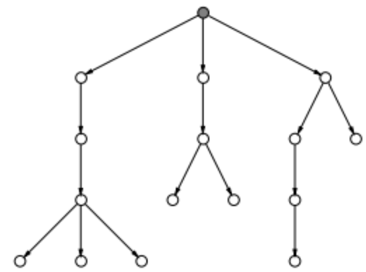
5 Cây

Cây là một đồ thị vô hướng liên thông không có chu trình. Một đỉnh bậc một trên cây gọi là **lá**. Một cây có N đỉnh thì sẽ có $N - 1$ cạnh. Giữa hai đỉnh tồn tại một và chỉ một đường đi đơn.



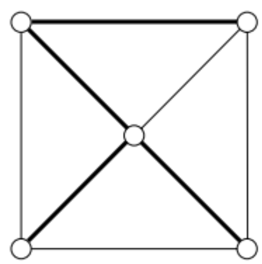
Hình 10: Đồ thị cây.

Cây có gốc là cây có một đỉnh gọi là **gốc** và các cạnh được định hướng theo đường đi đơn từ gốc tới các đỉnh khác. Nếu trên đường đi từ đỉnh gốc tới đỉnh y bất kì đi qua đỉnh x thì ta có thể gọi x là cha của y.



Hình 11: Cây có gốc.

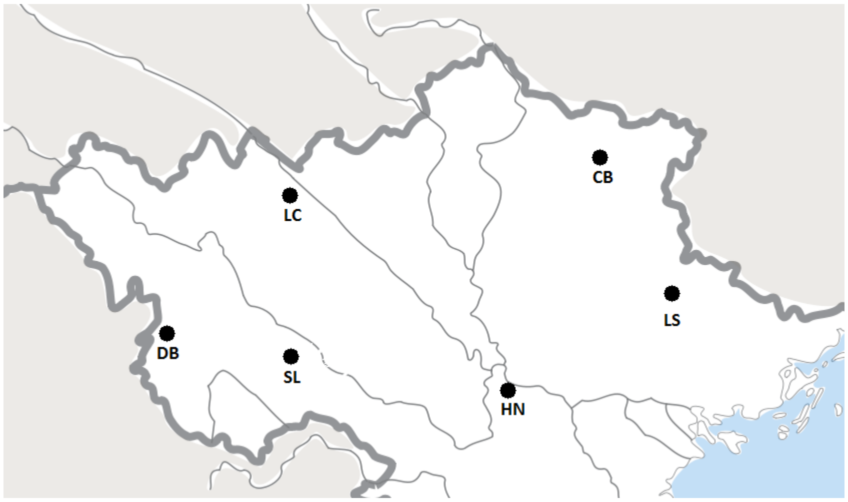
Cây khung một đồ vô hướng liên thông là một cây tạo ra bằng cách xoá một số cạnh trong đồ thị gốc.



Hình 12: Cây có hướng.

5.1 Cây khung nhỏ nhất

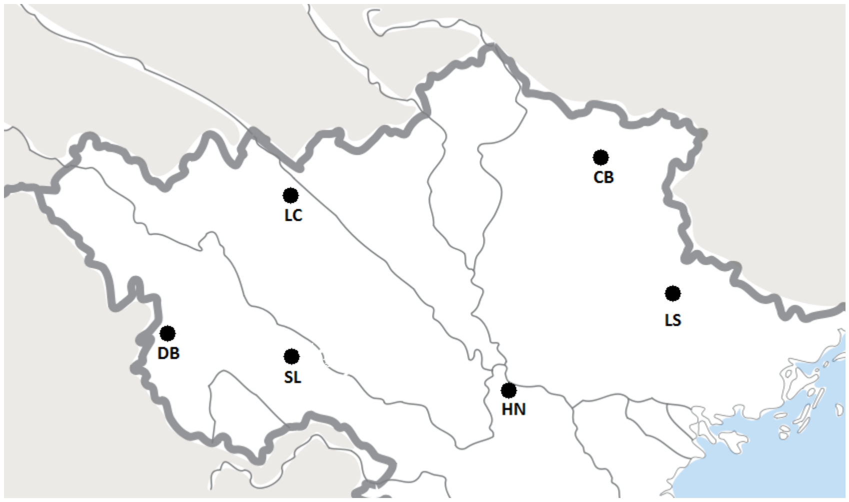
Các kĩ sư của PiMA cần thiết lập một hệ thống đường dây điện ở Việt Nam, kết nối các thành phố Hà Nội, Cao Bằng, Lạng Sơn, Lào Cai, Điện Biên đến Nhà máy thủy điện Sơn La. Vị trí và khoảng cách giữa các thành phố được thể hiện qua sơ đồ sau. Hỏi làm thế nào để xây dựng được một đường dây với tổng khoảng cách kết nối là ngắn nhất?



	HN	SL	CB	LS	DB	LC
HN	0	217	187	127	295	228
SL	–	0	311	312	101	144
CB	–	–	0	95	337	219
LS	–	–	–	0	364	261
DB	–	–	–	–	0	123
LC	–	–	–	–	–	0

Ta có thể đưa bài toán trên về dạng một đồ thị có trọng số với các đỉnh tương ứng với các thành phố (HN, SL, CB, LS, DB, LC). Ta cần tìm một đồ thị con bao trùm và liên thông với tổng các trọng số là nhỏ nhất. Bởi vì các trọng số tương ứng với khoảng cách giữa hai thành phố, là các số dương, nên đồ thị con cần tìm là một cây bao trùm. Bài toán trên là một trường hợp cu thể của bài toán tìm cây bao trùm có tổng trọng số nhỏ nhất trong một đồ thị liên thông có trọng số.

Trong thuật toán này, ta chọn một đỉnh bất kì làm gốc (root) của cây T. Gọi $\overline{E}(T)$ là tập hợp các cạnh có một đầu mút thuộc T nhưng bản thân cạnh đó không thuộc T. Ở mỗi bước, ta thêm vào T cạnh có trọng số nhỏ nhất trong $\overline{E}(T)$. Ta tô màu các đỉnh của T, và ở mỗi bước thì với mỗi đỉnh v chưa được tô màu, ta sẽ gán cho một "giá trị" $c(v)$, chính là trọng số nhỏ nhất của cạnh nối v với một đỉnh u đã được tô màu. Sau đó, ta chọn u làm "cha" của v và đặt hàm predecessor $p(v) = u$. Lưu ý rằng ban đầu mỗi đỉnh có giá trị là vô cùng và không có cha. Hai thông số p và c được cập nhật liên tục ở mỗi bước của thuật toán.



Hình 13: Bản đồ Việt Nam.

THUẬT TOÁN JARNIK-PRIM
input: một đồ thị liên thông có trọng số G.
output: một cây bao trùm của G có tổng các trọng số nhỏ nhất, hàm predecessor p, và "trọng lượng" của cây $w(T)$ (chính là tổng các trọng số).

1. cho $p(v) := \emptyset$ và $c(v) := \infty, v \in V$ và $w(T) := 0$
2. chọn một đỉnh r bất kì làm gốc
3. $c(r) := 0$
4. **while** có một đỉnh chưa được tô màu **do**
5. chọn đỉnh u sao cho $c(u)$ nhỏ nhất
6. tô màu u
7. **for** mỗi đỉnh v chưa được tô màu mà $w(uv) < c(v)$ **do**
8. $p(v) := u$ và $c(v) := w(uv)$
9. $w(T) := w(T) + c(u)$
10. **end for**
11. **end while**
12. return$(p, w(T))$

Hãy cùng áp dụng thuật toán Jarnik-Prim để giải quyết bài toán đường dây

điện nêu trên.
Bước 1, ta chọn Sơn La làm gốc, lúc này chưa có đỉnh nào được tô màu. Vì $c(SL) = 0$ và $c(v) = \infty$ với mọi $v \neq SL$, Sơn La chỉnh là đỉnh u được chọn ở bước 5. Tất cả các đỉnh khác nhận SL làm cha. Bây giờ ta lưu ý các giá trị của các đỉnh chưa được tô màu như sau:

$$c(HN) = 217,\ c(CB) = 311,\ c(LS) = 312,\ c(DB) = 101,\ c(LC) = 144$$

Lúc này $w(T)$ vẫn bằng 0.
Dựa vào các giá trị trên, ta sẽ chọn Điện Biên làm đỉnh để tô màu tiếp theo. Hàm predecessor và hàm giá trị thay đổi thành:

$$p(HN) = SL,\ p(CB) = SL,\ p(LS) = SL,\ p(LC) = DB$$

$$c(HN) = 217,\ c(CB) = 311,\ c(LS) = 312,\ c(LC) = 123$$

$$w(T) = 101$$

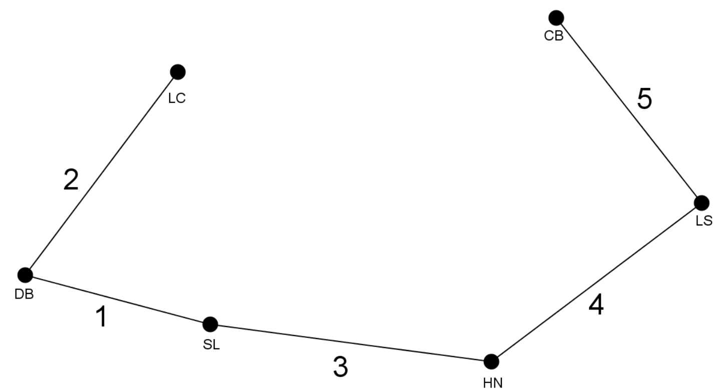
Ta thấy rằng $c(LC)$ là nhỏ nhất trong các giá trị trên, vậy ta chọn Lào Cai làm đỉnh tiếp theo được tô màu. Hàm predecessor và hàm giá trị trở thành:

$$p(HN) = SL,\ p(CB) = LC,\ p(LS) = LC$$

$$c(HN) = 217,\ c(CB) = 219,\ c(LS) = 261$$

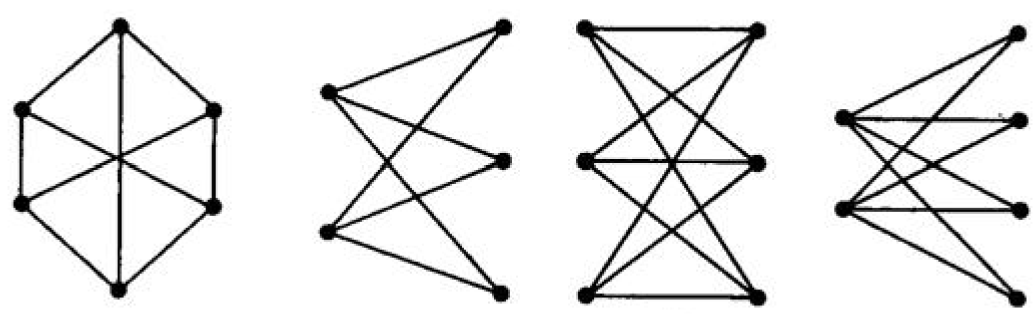
$$w(T) = 101 + 123 = 224$$

Cứ tiếp tục quá trình trên cho đến khi tất cả các đỉnh đều được tô màu. Cuối cùng ta sẽ thu được một cây bao trùm có tổng độ dài $w(T) = 663$km. Hình sau đây mô tả thứ tự được chọn của các cạnh và đỉnh.



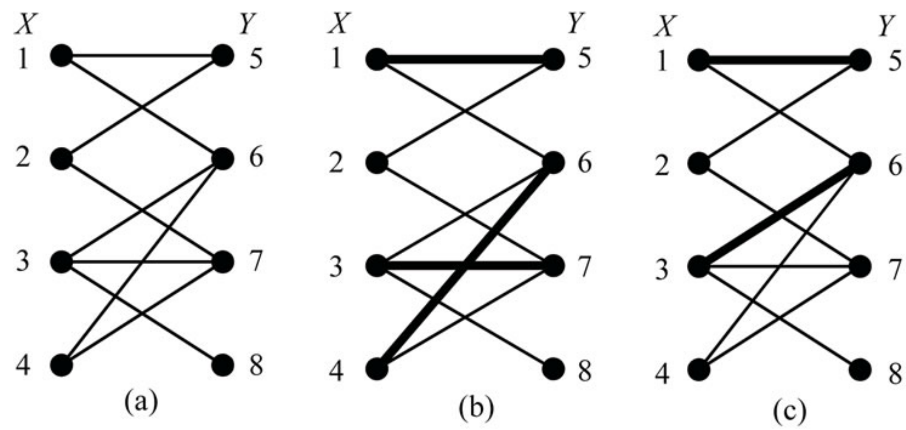
6 Đồ thị hai phía

Đồ thị hai phía là đồ thị tồn tại một cách phân tập đỉnh thành 2 tập X và Y sao cho không tồn tại cạnh nối u và v sao cho $u \in X$ và $v \in Y$. Đồ thị hai phía thường để biểu diễn quan hệ ghép cặp hoặc tương thích giữa hai tập đồ vật. Một đồ thị là đồ thị hai phía khi và chỉ khi đồ thị không tồn tại chu tình độ dài lẻ.



Hình 14: Đồ thị hai phía.

Một **bộ ghép** trong đồ thị hai phía là một tập con các cạnh của đồ thị sao cho không tồn tại một đỉnh của đồ thị liên thuộc quá 1 cạnh trong tập được chọn.



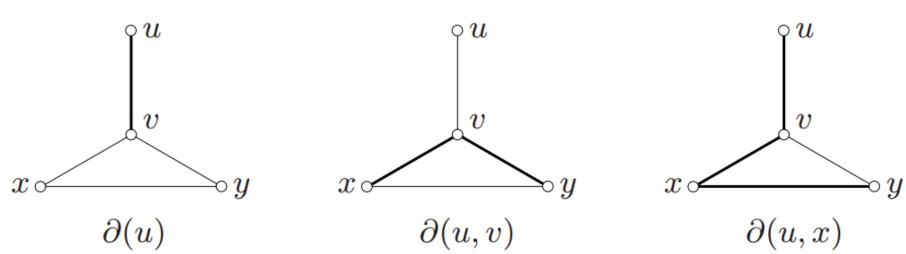
Hình 15: Bộ ghép trong đồ thị hai phía.

Mạng lưới

7 Một số định nghĩa và định lý quan trọng

7.1 Lát cắt cạnh

Cho đồ thị G và $X, Y \subset V(G)$ là các tập hợp đỉnh của G (X và Y không nhất thiết rời nhau). Ta kí hiệu $E[X, Y]$ là tập hợp các cạnh của G có một đỉnh thuộc X và đỉnh còn lại thuộc Y. Khi $Y = V(G) \setminus X$, tập hợp $E[X, Y]$ khi đó được gọi là một **lát cắt cạnh** của G ứng với X, và được kí hiệu bởi $\partial(X)$. Đặt $d(X) = |\partial(X)|$ thì $d(X)$ được gọi là **bậc** của tập hợp đỉnh X.



Trong đồ thị có hướng, ta dùng kí hiệu $A(X, Y)$ để chỉ tập hợp các cung có đỉnh đầu thuộc X và đỉnh cuối thuộc Y. Nếu $Y = V \setminus X$, ta gọi $A(X, Y)$ là **lát cắt ngoài** của D ứng với X và kí hiệu là $\partial^+(X)$. Tương tự như vậy,

$A(Y, X)$ được gọi là **lát cắt trong** ứng với X và được kí hiệu $\partial^-(X)$. Rõ ràng $\partial^+(X) = \partial^-(V \setminus X)$. $d^+(X) = |\partial^+(X)|$ được gọi là **bậc ngoài** của X. $d^-(X)$ được định nghĩa tương tự và được gọi là **bậc trong** của X. Nếu $X = \{x\}$ chỉ gồm một đỉnh thì ta chỉ kí hiệu đơn giản $d^+(x)$ và $d^=(x)$.

7.2 Mạng vận tải

Cho đồ thị có hướng D. Đỉnh $x \in A(D)$ được gọi là **đỉnh phát** nếu như $d^-(x) = 0$ (x có bậc trong bằng 0), và y được gọi là **đỉnh thu** nếu $d^+(y) = 0$. **mạng lưới** $N := N(x, y)$ là một đồ thị có hướng với đỉnh phát x và đỉnh thu y, cùng với một hàm số c xác định trên tập hợp cung của đồ thị. Các đỉnh khác x và y được gọi là **đỉnh trung gian**, và thường được kí hiệu bởi I (viết tắt của *intermediate vertices*). Hàm số c được gọi là **hàm năng suất** của N và với mỗi cung a thì giá trị $c(a)$ được gọi là **năng suất** (hay **khả năng thông qua**) của a.
Cấu trúc của mạng lưới trong lý thuyết đồ thị có nhiều nét tương đồng với mô hình vận chuyển hàng hóa trong cuộc sống, trong đó x tượng trưng cho nhà sản xuất, y là thị trường tiêu dùng và hàm c thể hiện năng suất vận chuyển.

7.3 Luồng

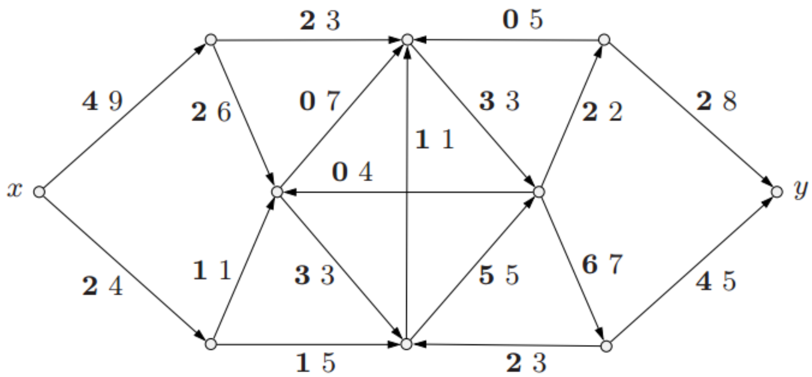
Cho f là một hàm số xác định trên tập hợp cung A của mạng lưới N. Với mỗi $S \subseteq A$, ta kí hiệu tổng $\sum\limits_{a \in S} f(a)$ bởi $f(S)$. Ngoài ra, với mỗi tập hợp đỉnh $X \subseteq V$, ta đặt

$$f^+(X) := f(\partial^+(X)) \text{ và } f^-(X) := f(\partial^-(X))$$

Một **luồng-**(x, y) (hay gọi tắt là **luồng**) trong N là một hàm số thực không âm f xác định trên A thỏa mãn:

1. $f^+(v) = f^-(v) \ \forall \ v \in I$
2. $0 \le f(a) \le c(a) \ \forall \ a \in A$

Ta có thể hiểu giá trị $f(a)$ là **dung lượng** của luồng qua cung a, và đẳng thức ở 1. nói rằng dung lượng luồng vào một đỉnh trung gian phải bằng dung lượng luồng ra khỏi đỉnh đó. Ngoài ra, một điều kiện quan trọng nữa là dung lượng luồng qua một cung không thể vượt quá năng suất của cung đó (điều kiện 2.). Mỗi mạng lưới đều tồn tại ít nhất một luồng, ví dụ luồng f xác định bởi $f(a) := 0$ với mọi a. Hình sau đây mô tả một luồng "không tầm thường", trong đó dung lượng của luồng qua mỗi cung được tô đậm bên cạnh năng suất của cung đó.



Nếu X là tập hợp một số đỉnh của N và f là một luồng trên N, thì ta gọi $f^+(X) - f^-(X)$ là *dung lượng luồng ra khỏi* X và $f^-(X) - f^+(X)$ là *dung lượng luồng đi vào* X. Không khó để thấy rằng, với mọi luồng-(x, y), dung lượng luồng ra khỏi x bằng dung lượng luồng đi vào y. Ta gọi giá trị của $f^+(x) - f^-(x)$ (hay $f^-(y) - f^+(y)$) là **tổng luồng từ** x **tới** y (hay đơn giản là **tổng luồng**), và được kí hiệu là (f) (viết tắt của *value*).
Ta có mệnh đề quan trọng sau đây thể hiện mối quan hệ giữa tổng luồng và một tập đỉnh X bất kì.

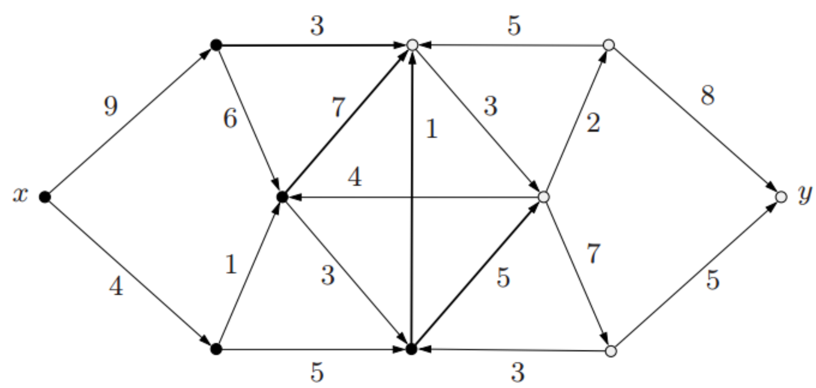
MỆNH ĐỀ 3.1
Với mọi luồng f trong mạng lưới $N(x, y)$ và bất kì tập đỉnh $X \subseteq V$ sao cho $x \in X$ và $y \in V \setminus X$, ta có

$$(f) = f^+(X) - f^-(X)$$

Luồng f được gọi là **luồng cực đại** nếu như không tồn tại một luồng nào khác có tổng luồng lớn hơn.

7.4 Lát cắt

Một **lát cắt** trong mạng lưới $N(x, y)$ là một lát cắt ngoài $\partial^+(X)$ sao cho $x \in X$ và $y \in V \setminus X$. **Năng suất** của một lát cắt $K := \partial^+(X)$ là tổng năng suất của các cung thuộc K, và được kí hiệu bởi (K). Trong hình dưới đây, các đường in đậm biểu diễn một lát cắt $\partial^+(X)$, trong đó X là các đỉnh được tô đậm. Năng suất của lát cắt trong hình là 3+7+1+5=16.



Với mỗi cung a, ta gọi a là *rỗng* nếu $f(a) = 0$, *dương* nếu $f(a) > 0$, *bão hòa* nếu $f(a) = c(a)$ và *chưa bão hòa* nếu $f(a) < c(a)$. Định lý sau đây cho ta một quan hệ hết sức quan trọng giữa tổng luồng và năng suất của một lát cắt bất kì.

ĐỊNH LÍ 3.2
Cho luồng f và một lát cắt bất kì $K := \partial^+(X)$ trong mạng lưới N. Khi đó:

$$(f) \leq (K)$$

Đẳng thức xảy ra khi và chỉ khi tất cả các cung trong K là bão hòa, và tất cả các cung trong $\partial^-(X)$ là rỗng.

Lát cắt K được gọi là **lát cắt cực tiểu** nếu như không tồn tại một lát cắt nào khác trong N có năng suất nhỏ hơn.

HỆ QUẢ 3.3
Cho f là một luồng và K là một lát cắt trong N. Nếu $(f) = (K)$ thì f là một luồng cực đại và K là một lát cắt cực tiểu.

7.5 Định lí Luồng cực đại-Lát cắt cực tiểu

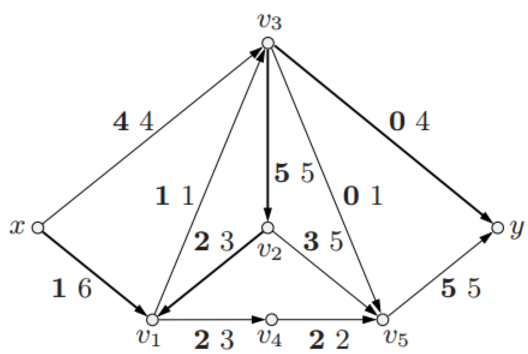
Cho f là một luồng trong $N(x, y)$ và một đường đi P bất kì trong N xuất phát từ x (đường đi này không nhất thiết phải là đường đi có hướng). Với mỗi cạnh $u_i u_{i+1}$ trong P tương ứng với cung a trong N, ta gọi a là *cung thuận* nếu a đi từ u_i tới u_{i+1}, và a là *cung nghịch* nếu ngược lại. Ta gán cho P một giá trị $\epsilon(P)$ định nghĩa như sau:

$$\epsilon(P) := \min\{\epsilon(a) : a \in A(P)\}$$

trong đó

$$\epsilon(a) := \begin{cases} c(a) - f(a) & \text{nếu } a \text{ là cung thuận} \\ f(a) & \text{nếu } a \text{ là cung nghịch} \end{cases}$$

Ý nghĩa của đại lượng $\epsilon(P)$ này là gì? Đây chính là giá trị lớn nhất mà ta có thể tăng vào dung lượng luồng qua P mà không làm ảnh hưởng đến hai điều kiện cơ bản của luồng. Đường đi P được gọi là **bão hòa** nếu $\epsilon(P) = 0$ và **chưa bão hòa** nếu $\epsilon(P) > 0$ (nói cách khác, tất cả cung thuận của P đều chưa bão hòa và tất cả cung nghịch đều dương). Nói một cách đơn giản, một đường đi chưa bão hòa thì chưa được sử dụng hết năng suất. **Đường đi tăng** là một đường đi chưa bão hòa từ x tới y. Trong ví dụ sau đây, đường đi $P := xv_1v_2v_3y$ là một đường tăng. Các cung thuận của P là xv_1 và v_3y, và $\epsilon(P) = \min\{5, 2, 5, 4\} = 2$.



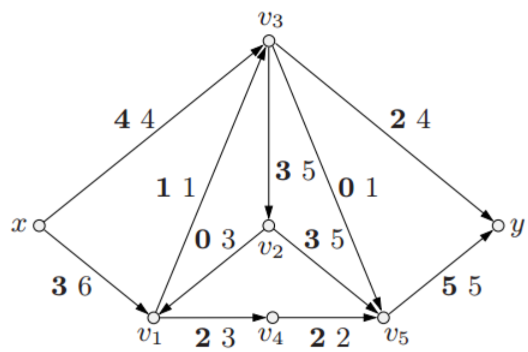
Đường đi tăng trong một mạng lưới có vai trò rất quan trọng vì nó cho ta biết liệu f có phải là một luồng cực đại hay không. Bằng cách tăng thêm $\epsilon(P)$ vào luồng đi qua P, ta có thể thu được một luồng mới f' với tổng luồng lớn hơn. Cụ thể, xác định f' như sau:

$$f'(a) := \begin{cases} f(a) + \epsilon(P) & \text{nếu } a \text{ là cung thuận} \\ f(a) - \epsilon(P) & \text{nếu } a \text{ là cung nghịch} \\ f(a) & \text{nếu } a \text{ không thuộc } P \end{cases}$$

Ta có mệnh đề sau đây.

MỆNH ĐỀ 3.4
Cho f là một luồng trong mạng lưới N. Nếu tồn tại một đường đi tăng P, khi đó f không phải là luồng cực đại. Cụ thể, luồng f' xác định như trên có tổng luồng $(f') = (f) + \epsilon(P) > (f)$.

Trong đồ thị minh họa ở trên, sau khi biến đổi luồng f thành luồng f' thì mạng lưới thu được sẽ là:



Điều gì sẽ xảy ra nếu như trong mạng lưới không có đường đi tăng? Mệnh đề sau đây sẽ giải quyết vấn đề này.

MỆNH ĐỀ 3.5
Cho f là một luồng trong mạng lưới $N(x, y)$. Giả sử không tồn tại đường đi tăng trong N. Gọi X là tập hợp các đỉnh tới được từ x bởi một đường đi chưa bão hòa, và đặt $K := \partial^+(X)$. Khi đó f là luồng cực đại và K là lát cắt cực tiểu.

Hệ quả quan trọng của mệnh đề 3.4 và mệnh đề 3.5 là định lí sau đây, được phát hiện độc lập bởi Elias et al. (1956) và Ford-Fulkerson (1956).

ĐỊNH LÍ LUỒNG CỰC ĐẠI-LÁT CÁT CỰC TIỂU
Trong một mạng lưới bất kì, tổng luồng của luồng cực đại bằng với năng suất của lát cắt cực tiểu.

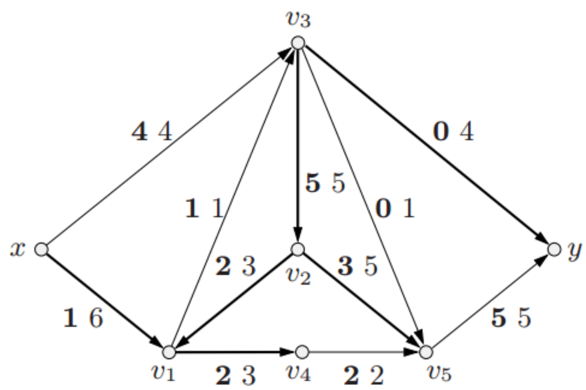
8 Bài toán tìm cặp ghép lớn nhất

8.1 Thuật toán Fork-Fulkerson

Định li sau đây là hệ quả trực tiếp từ mệnh đề 3.4 và mệnh đề 3.5.

ĐINH LÍ 3.6
Trong mạng lưới, một luồng là cực đại nếu và chỉ nếu không tồn tại đường đi tăng ứng với luồng đó.

Định lí này là tiền đề cho thuật toán tìm luồng cực đại trong một mạng lưới. Bắt đầu với một luồng f bất kì cho trước, chẳng hạn luồng 0, ta tìm một đường đi tăng bằng một thuật toán tìm cây. Một cây có gốc x T được gọi là *cây chưa bão hòa* nếu như với mọi $v \in T$, đường đi xTv là chưa bão hòa (đường đi vô hướng). Sau đây là ví dụ về một cây chưa bão hòa (các cạnh được tô đậm).

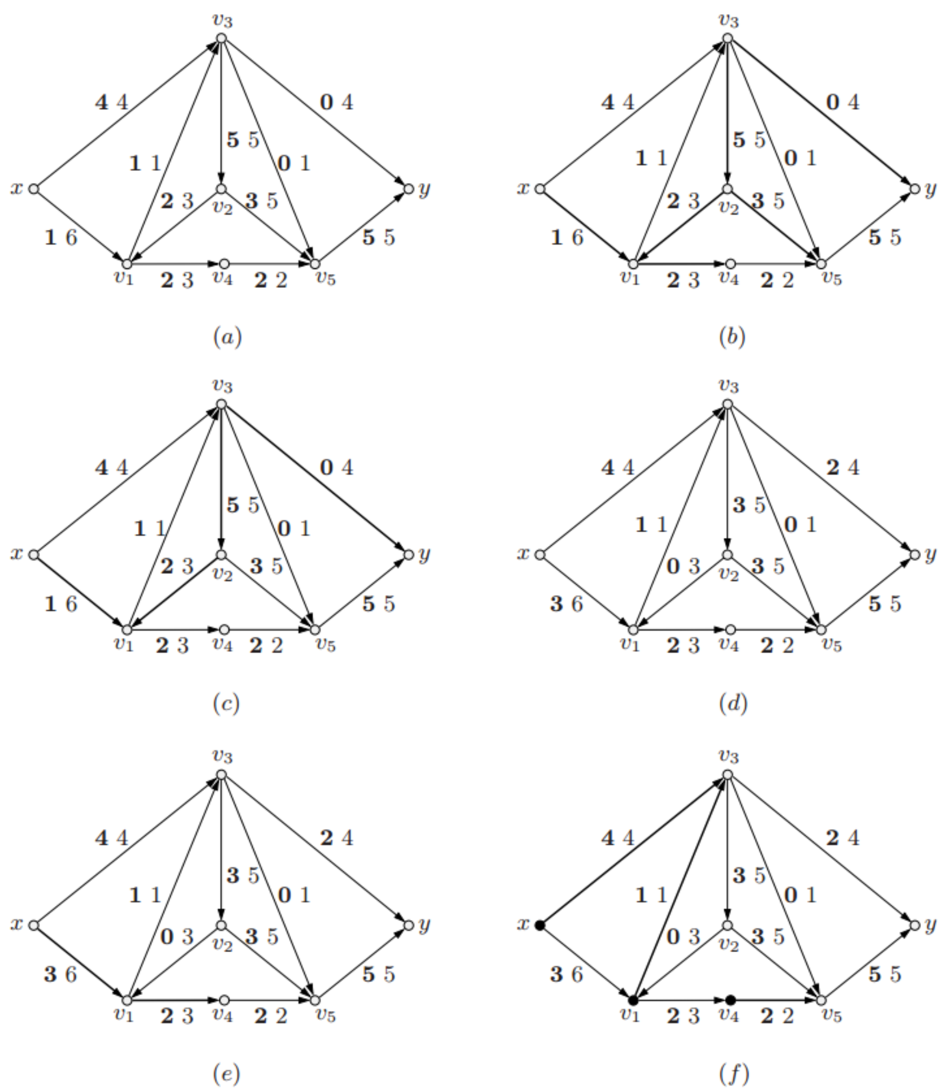


Bắt đầu thuật toán, cây T chỉ bao gồm duy nhất một đỉnh x. Tại mỗi bước, có hai cách ta có thể mở rộng cây này. Nếu tồn tại một cung chưa bão hòa a trong $\partial^+(X)$, với $X = V(T)$, thì ta thêm cả a và đỉnh cuối của a vào T (đỉnh đầu của a đã thuộc T vì $a \in \partial^+(X)$. Tương tự, nếu có một cung dương a trong $\partial^-(X)$ thì ta cũng thêm cả a và đỉnh cuối của a vào T. Đến cuối quá trình, nếu T đến được y (y thuộc T) thì xTy là một đường đi tăng, và ta thay f bởi f' xác định như trên. Nếu quá trình dừng lại mà T không đến được y, thì T là một cây chưa bão hòa cực đại, và mỗi cung trong $\partial^+(X)$ đều là cung bão hòa, cũng như mỗi cung trong $\partial^-(X)$ là cung rỗng (chú ý ta vẫn dùng X để chỉ tập đỉnh $V(T)$ của T). Đến đây ta có thể kết luận từ định lí 3.2 rằng f là luồng cực đại và $\partial^+(X)$ là lát cắt cực tiểu.

THUẬT TOÁN FORD-FULKERSON
input: mạng lưới $N := N(x, y)$ và luồng f trong N.
output: luồng f cực đại và lát cắt $\partial^+(X)$ cực tiểu trong N.

1. cho $X := \{x\}, p(v) := \emptyset, v \in V$
2. **while** có một cung chưa bão hòa $a := uv$ hoặc một cung dương $a := uv$ với $u \in X$ và $v \in V \setminus X$ **do**
3. $X := X \cup \{v\}$
4. $p(v) := u$
5. **end while**
6. **if** $y \in X$ **then**
7. tính $\epsilon(P) := \min\{\epsilon(a) : a \in A(P)\}$, trong đó P là đường đi xy trong cây vừa tìm ở trên với hàm predecessor p
8. với mỗi cung thuận a của P, $f(a) := f(a) + \epsilon(P)$

9. với mỗi cung nghịch a của P, $f(a) := f(a) - \epsilon(P)$
10. trở lại 1
11. **end if**
12. return$(f, \partial^+(X))$



(a) luồng f, (b) cây chưa bão hòa, (c) đường đi tăng, (d) đường đi sau khi thay đổi, (e) cây chưa bão hòa cực đại không chứa y, (f) lát cắt cực tiểu

LƯU Ý: Nếu tất cả giá trị của năng suất đều nguyên hoặc hữu tỉ thì thuật toán sẽ dừng sao hữu hạn bước. Tuy nhiên, điều đó chưa chắc đúng nếu có năng suất nhận giá trị vô tỉ.

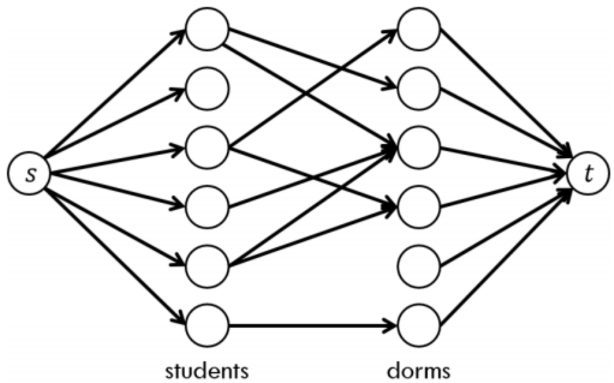
8.2 Bài toán tìm cặp ghép lớn nhất

Năm 2015, có 2500 học sinh trên toàn thế giới được nhận vào trường Đại học Chicago (The University of Chicago). Trường có tổng cộng 9 kí túc xá, và mỗi học sinh được đăng kí với trường kí túc xá mà mình muốn ở. Bảng sau đây cho ta biết lượng học sinh tối đa mà mỗi kí túc xá có thể chứa. Hãy tìm một cách sắp xếp sao cho có nhiều học sinh được vào kí túc xá mình đăng kí nhất có thể.

Blackstone	120
Broadview	145
International House	500
Max Palevsky	200
Snell - Hitchcock	175
Burton - Judson	210
Stony Island	280
Renee Granville - Grossman	270
Campus North	600

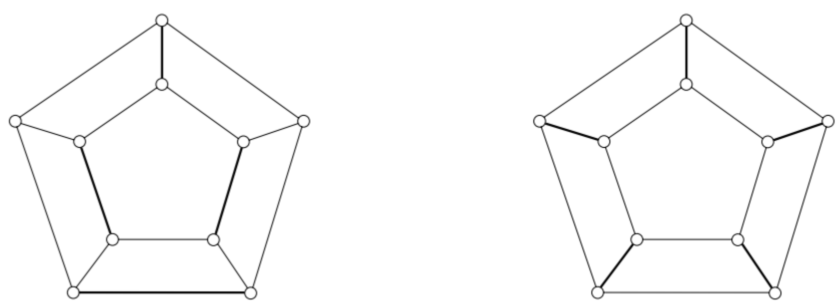
Một cách tự nhiên, ta sẽ nghĩ đến mô hình đồ thị bao gồm các đỉnh biểu diễn học sinh và kí túc xá, và cạnh nối học sinh x với kí túc xá y nếu và chỉ nếu x chọn y làm nơi mình muốn ở. Tuy nhiên, mô hình này vẫn là một đồ thị vô hướng, và để tìm cách sắp xếp học sinh-kí túc xá tối ưu là tương đối phức tạp. Ta sẽ chuyển mô hình này thành một mạng lưới như sau:

- Thêm đỉnh phát s và đỉnh thu t.
- Nối s đến tất cả các đỉnh biểu diễn học sinh sao cho s là đỉnh đầu, đồng thời tất cả các cung sx, với x là học sinh bất kì, có năng suất 1.
- Nối t đến tất cả các đỉnh biểu diễn kí túc xá sao cho t là đỉnh cuối, đồng thời tất cả các cung yt, với y là kí túc xá bất kì, có năng suất 1.
- Với mỗi học sinh x chọn kí túc xá y, ta cho x là đỉnh đầu và y là đỉnh cuối, đồng thời cung xy có năng suất 1.



Bằng cách đưa về mô hình này, ta đã đưa bài toán ban đầu về bài toán tìm luồng lớn nhất của mạng lưới $N(s,t)$ như trên (tại sao?), và hoàn toàn có thể giải được bằng thuật toán Ford-Fulkerson.
Bài toán trên là ví dụ điển hình của một ứng dụng quan trọng của mạng lưới: tìm cặp ghép lớn nhất trong đồ thị hai phe.

Cặp ghép hoàn hảo là cặp ghép phủ hết tất cả các đỉnh. Nói cách khác, mỗi đỉnh của đồ thị thuộc đúng một cạnh trong cặp ghép hoàn hảo.
Cặp ghép lớn nhất là cặp ghép chứa nhiều cạnh nhất có thể. **Cặp ghép tối đa** M là cặp ghép mà nếu ta thêm một cạnh chưa thuộc M vào M thì tập hợp cạnh thu được không còn là một cặp ghép. Rõ ràng một cặp ghép lớn nhất thì tối đa, nhưng điều ngược lại chưa chắc đúng. Ví dụ say đây chứng tỏ điều này.



Hiển nhiên, nếu tồn tại một cặp ghép hoàn hảo thì nó cũng là cặp ghép lớn nhất (như hình bên trái của ví dụ trên).
Từ những định nghĩa trên, ta có thể tổng quát hóa bài toán học sinh-kí túc xá trên thành bài toán tìm cặp ghép lớn nhất trong một đồ thị hai phe $G[X,Y]$. Cách làm cũng hoàn toàn tương tự, ta có thể thêm vào đỉnh phát và đỉnh thu rồi chuyển đồ thị vô hướng thành đồ thị có hướng với các cạnh có năng suất 1.