CHƯƠNG 5

LUỒNG TRONG MẠNG
(NETWORK FLOWS)

1. GIỚI THIỆU

Khắp nơi trong đời sống hàng ngày của chúng ta hiện diện các loại mạng (network) khác nhau. Các mạng lưới điện và năng lượng mang lại ánh sáng và các phương tiện giải trí cho chúng ta. Mạng điện thoại cho chúng ta khả năng liên lạc với nhau dù đang ở nơi nào trên thế giới. Hệ thống đường quốc lộ, đường sắt và đường hàng không cho phép chúng ta vượt qua những khoảng cách địa lý khác nhau để thực hiện công tác của mình hay đi du lịch, giải trí sau những ngày làm việc nặng nề. Mạng lưới sản xuất và phân phối sản phẩm cung cấp cho chúng ta thực phẩm và các nhu yếu phẩm thường ngày. Và cuối cùng, mạng lưới máy tính như mạng đặt vé máy bay, mạng ngân hàng, nhất là internet đã thay đổi về chất phương thức trao đổi thông tin, cách thực hiện các thương vụ và cả đời sống hàng ngày của chúng ta.

Trong tất cả các lãnh vực đề cập ở trên, và trong rất nhiều lãnh vực khác nữa, chúng ta muốn chuyển một thực thể nào đó (dòng điện, sản phẩm, con người hay xe cộ hoặc các thông điệp, ...) từ một điểm đến một vị trí khác trong một mạng lưới tương ứng, và ta muốn thực hiện công việc này sao cho hiệu quả nhất có thể.

Luồng trong mạng là một lớp các bài toán liên quan đến nhiều lãnh vực khác nhau như toán ứng dụng, tin học, công nghệ, quản lý, quản trị, ... Bài toán luồng trong mạng có lịch sử phát triển khá lâu đời kể từ khi những công trình đầu tiên liên quan đến lãnh vực này được công bố bởi Gustav Kirchhof và các nhà tiên phong khác. Họ là những người đầu tiên phân tích một cách có hệ thống về dòng điện. Họ đã sử dụng mạng (network hay đồ thị) như những phương tiện hữu ích biểu diễn nhiều hệ thống vật lý khác nhau.

Ngoài ra, chúng ta cũng sẽ xem xét đến một số vấn đề không hoàn toàn thuộc về lớp các bài toán luồng trên mạng nhưng có nhiều liên quan.

2. BÀI TOÁN LUỒNG TRONG MẠNG

Trước tiên, chúng ta cùng nhau xem xét một số mô hình luông trong mạng. Sau đó ở các phần sau, chúng ta sẽ cùng nhau xem xét một vài ứng dụng của luồng trong mạng.

2.1. Bài toán luồng với chi phí cực tiểu (Minimum cost flow problem)

Mô hình luồng với chi phí cực tiểu (gọi tắt là luồng cực tiểu) là mô hình cơ sở của tất các bài toán luồng trong mạng. Bài toán có thể phát biểu như sau: *Ta muốn xác định chi phí thấp nhất cho một chuyến vận chuyển hàng (shipment) theo yêu cầu từ nơi cung cấp đến chỗ đặt hàng.* Mô hình này có một số ứng dụng gần gũi như: phân phối sản phẩm từ các chi nhánh sản xuất đến các kho chứa hoặc từ kho chứa đến các nhà bán lẻ; luồng luân chuyển các nguyên vật liệu qua các tổ hợp máy trong một dây chuyền sản xuất; luồng di chuyển của các ô tô trong một mạng lưới giao thông đô thị; đường đi của các cuộc gọi trong mạng điện thoại …

Mô hình toán học của bài toán luồng cực tiểu như sau:

Cho G=(N, A) là một đồ thị (mạng) có hướng định nghĩa bởi tập hợp N gồm n nút và tập hợp A gồm m cung. Mỗi cung (i,j) $\in$ A được gán một giá trị c_{ij} gọi là chi phí (cost) của cung (i,j). Nó cho biết chi phí của một luồng đơn vị chuyển tải qua cung (i,j). Ta giả thiết chi phí của luồng thay đổi một cách tuyến tính theo kích thước của nó. Ta cũng gán cho mỗi cung (i,j) hai giá trị u_{ij} và l_{ij} gọi là khả năng thông qua của cung (i,j). Giá trị u_{ij} và l_{ij} lần lượt cho biết khối lượng luồng tối đa và tối thiểu có thể (phải) chuyển tải qua cung. Mỗi một nút $i \in N$ được gán một giá trị b(i) biểu diễn khả năng cung/cầu của nút. Nếu $b(i) > 0$, nút i gọi là nút cung (nút nguồn); nếu $b(i) < 0$, nút i gọi là nút cầu (nút đích); và nếu $b(i) = 0$, nút i gọi là nút trung chuyển. Các ẩn số cần xác định trong bài toán luồng cực tiểu là giá trị luồng tại các cung (ta ký hiệu là x_{ij}). Bài toán luồng cực tiểu là một bài toán tối ưu được mô hình như sau:

Cực tiểu hóa giá trị $\sum_{(i,j)\in A} c_{ij} x_{ij}$

Với các ràng buộc:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ij} = b(i), \forall i \in N,$$

$$l_{ij} \le x_{ij} \le u_{ij}, \forall (i,j) \in A,$$

trong đó, $\sum_{i=1}^{n} b(i) = 0$.

Dưới dạng ma trận, ta có thể biểu diễn bài toán luồng cực tiểu như sau:

Cực tiểu hóa **cx**

Với các ràng buộc:

$$\aleph \mathbf{x} = \mathbf{b}$$

$$\mathbf{l} \le \mathbf{x} \le \mathbf{u}.$$

Trong cách biểu diễn này, $\aleph$ là một ma trận n×m, gọi là *ma trận liên thuộc nút-cung (node-arc incidence matrix)* của bài toán luồng cực tiểu. Mỗi cột $\aleph_{(i,j)}$ (ứng với cung (i,j)) có giá trị +1 tại dòng i và –1 tại cột j. Các phần tử khác của cột đều bằng 0.

Ta gọi ràng buộc (1.b) là ràng buộc về cân bằng vật chất (mass balance contraint). Tổng đầu tiên trong ràng buộc này cho biết tổng giá trị luồng đi ra từ một nút và tổng thứ 2 cho biết tổng giá trị luồng đi vào nút đó. Ràng buộc về cân bằng vật chất bắt buộc hiệu của tổng luồng vào và ra của một nút phải bằng khả năng cung/cầu của nó. Các giá trị luồng còn phải thỏa mãn chặn trên u_{ij} và chặn dưới l_{ij} của cung. Trong đa số ứng dụng, già trị của l_{ij} thường là 0. Vì vậy, nếu ta không nhắc tới giá trị l_{ij} trong một ứng dụng nào đó thì ta phải hiểu rằng chúng có giá trị ngầm định là 0.

Trong đa số trường hợp, vì lý do đơn giản, ta sẽ giả thiết các giá trị trong bài toán luồng cực tiểu là các số nguyên. Giả thiết này sẽ không làm mất tính tổng quát của bài toán. Sau đây là một số trường hợp đặc biệt của bài toán luồng cực tiểu:

2.2. Bài toán tìm đường đi ngắn nhất

Bài toán này chúng ta đã khảo sát kỹ trong phần trước của quyển sách này. Ở đây chúng ta sẽ phát biểu nó như một trường hợp đặc biệt của bài toán luồng trên mạng.

Bài toán đường đi ngắn nhất có lẽ là bài toán dễ nhất trong các bài toán luồng trong mạng. Đối với bài toán này, chúng ta mong muốn tìm được một đường đi với chi phí thấp nhất từ một nút nguồn s tới nút đích t, giả sử rằng mỗi cung $(i,j)\in A$ có chi phí tương ứng là c_{ij}. Nếu ta đặt b(s) = 1, b(t)=-1 và b(i) = 0 với mọi nút i còn lại, lời giải của bài toán tìm đường đi ngắn nhất từ s đến t chính là lời giải của bài toán luồng cực tiểu khi ta gửi 1 đơn vị luồng từ s đến t (sẽ đi theo đđnn). Nếu ta đặt b(s) = n-1 và b(i) = 0 với mọi nút i còn lại, ta sẽ nhận được mô hình luồng cực tiểu của bài toán tìm đường đi ngắn nhất từ s đến tất cả các đỉnh còn lại.

2.3. Bài toán luồng cực đại (maximum flow problem)

Bài toán luồng cực đại tìm một phương án phù hợp để gửi 1 luồng có giá trị lớn nhất có thể từ nút nguồn s đến nút đích t. Nếu khả năng thông qua cực đại của cung (i,j) là u_{ij}, bài toán luồng cực đại có thể biểu diễn như trường hợp riêng của bài toán luồng cực tiểu nếu ta đặt $b(i) = 0\ \forall i \in N$, $c_{ij} = 0\ \forall (i,j) \in A$, và thêm vào mạng cung (t,s) với c_{ts}=-1 và $u_{ts}= \infty$.

Lời giải của bài toán luồng cực tiểu sẽ cho ta giá trị cực đại của luồng trên cung (t,s); nhưng vì luồng bất kỳ trên cung (t,s) sẽ chuyển ngược từ nút s đến nút t thông qua các cung thuộc A (bởi vì b(i) = 0). Vì vậy, lời giải của bài toán luồng cực tiểu sẽ chính là giá trị luồng cực đại trên mạng gốc ban đầu.

Luồng cực đại sẽ là một trong những bài toán trung tâm của chương này.

2.4. Bài toán phân công (assignment problem)

Trong bài toán phân công, ta có 2 tập hợp cùng kích thước N_1 và N_2 (nghĩa là $|N_1| = |N_2|$), một tập hợp các cặp $A \subseteq N_1 \times N_2$ biểu diễn các khả năng phân công, và chi phí phân công c_{ij} ứng với mỗi phần tử $(i,j) \in A$. Trong bài toán phân công, ta cần ghép cặp mỗi phần tử i trong N_1 với một phần tử j trong N_2 một cách duy nhất (song ánh) sao cho tổng chi phí phân công là thấp nhất. Các bài toán phân công thường gặp trong thực tế rất nhiều. Chẳng hạn như bài toán phân công người tham gia các dự án, phân công công việc thực hiện trên các máy, ... Bài toán phân công chính là bài toán luồng cực tiểu trong mạng (đồ thị) 2 phía $G=(N_1 \cup N_2, A)$ với $b(i) = 1\ \forall i \in N_1$, $b(i) = 1\ \forall i \in N_2$ và $u_{ij} = 1\ \forall (i,j) \in A$.

2.5. Bài toán vận tải (transportation problem)

Bài toán vận tải là trường hợp riêng của bài toán luồng cực tiểu với các đặc tính như sau: tập nút N được chia làm 2 tập con N_1 và N_2 sao cho (1) mỗi nút trong N_1 là nút cung, (2) mỗi nút trong N_2 là nút cầu và (3) với mọi cung $(i,j) \in A$, $i \in N_1$ và $j \in N_2$. Trong bài toán này, thường $l_{ij} = 0$. Ví dụ kinh điển về bài toán mạng vận tải chính là bài toán phân phối hàng từ các kho chứa đến khách hàng. Trong ví dụ này, N_1 biểu diễn tập các kho chứa, N_2 biểu diễn tập các khách hàng và mỗi cung $(i,j) \in A$ biểu diễn một kênh phân phối từ kho i đến khách hàng j.

2.6. Bài toán lưu thông (circulation problem)

Bài toán lưu thông là bài toán luồng cực tiểu mà trong đó mạng chỉ chứa các nút trung gian (nghĩa là $b(i) = 0\ \forall i \in N$). Trong bài toán này, ta muốn tìm một luồng tương thích thỏa mãn chặn dưới l_{ij} và chặn trên u_{ij} của giá trị luồng x_{ij} trên mọi cung (i,j). Do trong mạng không có nút cung cũng như nút cầu nên ta không thêm vào cũng như không lấy gì ra khỏi luồng, toàn bộ luồng sẽ luân chuyển trong mạng. Mục tiêu của bài toán là tìm một luồng tương thích với chi phí tối thiểu. Bài toán thiết kế các tuyến bay của một hãng hàng không là một ví dụ của bài toán lưu thông. Trong ví dụ này, mọi máy bay của hãng lưu thông trong một mạng lưới các thành phố. Mỗi cung (i,j) có giá trị luồng chặn dưới $l_{ij} = 1$ nếu hãng hàng không cần cung cấp tuyến bay từ thành phố i đến thành phố j. Lúc đó, hãng cần điều phối ít nhất 1 máy bay cho tuyến đường này.

2.7. Bài toán luồng tổng quát (generalized flow problem)

Trong bài toán luồng cực tiểu, các cung bảo toàn giá trị luồng. Nghĩa là, giá trị luồng đi vào một cung bằng giá trị luồng đi ra từ cung đó. Trong bài toán luồng tổng quát, một cung có thể làm giảm hoặc tăng giá trị luồng với một hệ số nhất định. Nếu giá trị luồng vào cung (i, j) là x_{ij} thì luồng ra từ cung này sẽ có giá trị $\mu_{ij}x_{ij}$. Nếu $0 < \mu_{ij} < 1$ thì cung (i, j) làm mất mát luồng, còn nếu $1 < \mu_{ij} < \infty$ thì cung (i, j) sẽ sinh thêm luồng. Bài toán luồng tổng quát gặp trong nhiều ứng dụng thực tế khác nhau. Chẳng hạn như (1) bài toán truyền tải năng lượng điện, trong đó năng lượng điện sẽ bị hao tổn trên đường truyền, (2) mạng cung cấp nước với việc hao tổn nước do sự rò rỉ, (3) bài toán vận chuyển hàng hóa có sự hao hụt (rau, quả, nước đá, ...) và (4) vấn đề luân chuyển và đầu tư tiền mặt trong đó mỗi cung ứng với khả năng đầu tư và hệ số μ_{ij} biểu diễn sự tăng giá hoặc giảm giá của giá trị đầu tư.

2.8. Bài toán cặp ghép (matching problem)

Bài toán cặp ghép tuy không thuộc mô hình bài toán luồng trên mạng nhưng nó là một mô hình các bài toán rất quan trong trên mạng. Tuy về mặt toán học bài toán cặp ghép có mô hình rất khác so với mô hình luồng trên mạng nhưng bài toán này có những mối liên hệ mật thiết với một số bài toán luồng trên mạng.

Một bộ cặp ghép trên đồ thị G=(N, A) là một tập hợp các cung với thuộc tính sau: mỗi nút kề với tối đa một cung trong tập hợp. Như vậy, trong một bộ cặp ghép, mỗi nút được ghép cặp với tối đa một nút khác. Bài toán cặp ghép sẽ tìm một bộ cặp ghép theo một tiêu chuẩn tối ưu nào đó. Bài toán cặp ghép trên đồ thị hai phía tìm một bộ cặp ghép từ hai tập hợp nút.

3. MỘT SỐ ỨNG DỤNG CỦA BÀI TOÁN LUỒNG CỰC TIỂU

Có thể nói, mạng xuất hiện khắp nơi. Có những bài toán, nhìn vào ta thấy ngay sự hiện diện của mạng, nhưng cũng có những bài toán mà mạng bị che khuất bởi phát biểu của nó. Để ứng dung được các kết quả nghiên cứu về luồng trong mạng, ta cần thiết lập được mạng tương ứng trong các bài toán này. Ứng dụng kinh điển của bài toán luồng trong mạng là nghiên cứu vầ bài toán vận tải. Trong bài toán này, nhà cung cấp với các thông tin về hàng tồn kho tại các kho chứa của mình phải vận chuyển hàng hóa từ các kho này đến các đại lý bán lẻ của mình nằm ở những vị trí địa lý rất khác nhau. Mỗi đại lý đều có những yêu cầu cụ thể về các loại hàng hóa tùy theo đòi hỏi của các khách hàng của mình. Và đương nhiên, nhà cung cấp muốn tối thiểu hóa chi phí vận chuyển.

Bảng dưới đây sẽ liệt kê một số mô hình mạng thường gặp:

Ứng dụng	**Nút**	**Cung**	**Luồng**
Các hệ thống truyền tin	Máy điện thoại, máy tính, vệ tinh, các trạm truyền tin, ...	Dây cáp, cáp quang, kênh liên kết vô tuyến,...	Lời nói, dữ liệu, tín hiệu video,...
Các hệ thống cấp nước, chất lỏng,...	Trạm bơm, hồ chứa, hồ, ...	Ống dẫn	Nước, ga, dầu, ...
Mạch IC	Các cổng, thanh ghi, các bộ vi xử lý, ...	Dây dẫn	Dòng điện
Các hệ thống cơ khí	Các khớp nối	Cần, thanh nối, lò xo, con lắc, ...	Sức nóng, năng lượng, ...
Mạng giao thông	Giao lộ, sân bay, nhà ga, ...	Con đường, tuyến bay, đường ray xe lửa, ...	Hàng khách, hàng hóa, xe cộ, ...

3.1. Ứng dụng 1: *Chuyển nhà*

Một chủ nhà có một số căn nhà muốn cho thuê. Mỗi căn nhà có những đặc điểm riêng. Ví dụ, một căn nhà có thể có hay không garage, có một số lượng phòng ngủ, phòng tắm và có giá cho thuê khác nhau. Các tham số này sẽ giúp chúng ta nhóm các căn nhà thành những loại khác nhau được đánh số 1, 2, ..., n.

Sau một thời gian, một số khách thuê sẽ trả lại căn nhà đang thuê và chuyển đến ở chỗ mới phù hợp với nhu cầu của mình hơn. Như vậy, đòi hỏi của khách thuê thay đổi theo thời gian. Chủ nhà đương nhiên muốn chuyển những người khách thuê đến căn nhà mới phù hợp với nhu cầu và khả năng mới của họ. Có thể xảy ra trường hợp xoay vòng khi a chuyển đến nhà b, b chuyển đến nhà c, ..., d chuyển đến nhà a. Bài toán đặt ra là tìm xem liệu có một chu trình như vậy tồn tại hay không.

Để giải bài toán này như là một bài toán trên mạng, trước tiên ta tạo một đồ thị G trong đó tập hợp các nút biểu diễn các loại nhà khác nhau. Ta thêm một cung (i,j) vào đồ thị khi một người đang sống trong căn nhà thuộc loại i muốn chuyển sang căn nhà loại j. Một chu trình trong G biểu diễn chu trình thay đổi nhà thỏa mãn đòi hỏi của tất cả những người đang sống tại những căn nhà tham gia vào chu trình. Lặp đi lặp lại phương pháp này ta có thể thỏa mãn đòi hỏi của ngày càng nhiều khách hàng hơn.

Ứng dụng này đòi hỏi một phương pháp xác định chu trình trong mạng nếu nó tồn tại. Một phương pháp được nhiều người biết là Topo Sort. Nói chung, có nhiều cách di chuyển khách thuê bởi vì đồ thị G có thể chứa nhiều chu trình khác nhau. Trong trường hợp này chủ nhà mong muốn tìm một chu trình chứa càng ít cung càng tốt bởi vì càng ít sự di chuyển thì càng dễ quản lý khi thay đổi. Ta có thể giải quyết vấn đề này bằng cách giải bài toán tìm đường đi ngắn nhất.

3.2. Ứng dụng 2: *Sự phân loại những sợi thép*

Trong những dự án xây dựng khác nhau, một công ty xây dựng cần những sợi thép có cấu trúc của mặt cắt giống nhau nhưng có chiều dài khác nhau. Với mỗi i = 1..n cho $D_i>0$ biểu diễn yêu cầu số lượng sợi thép có chiều dài L_i và giả sử rằng $L_1<L_2<...<L_n$. Công ty có thể đáp ứng những yêu cầu bằng cách lưu trữ và cung cấp đúng D_i sợi thép có chiều dài L_i, $\forall i$. Tuy

nhiên, có thể không kinh tế nếu lấy đúng số lượng tất cả các sợi thép có chiều dài được yêu cầu vì tốn nhiều chi phí để lưu trữ và quản lý tất cả các loại sợi thép với chiều dài khác nhau. Trong trường hợp nếu công ty cần một sợi thép có chiều dài L_i mà không có trong kho, công ty có thể cắt một sợi thép dài hơn thành chiều dài cần lấy. Việc cắt các sợi thép thường phát sinh ra các đoạn thừa được coi là phế liệu. Gọi K_i là chi phí để tổ chức quản lý các sợi thép có chiều dài L_i và C_i là giá của sợi thép này. Công ty muốn xác định cụ thể số lượng của từng loại sợi thép để đáp ứng yêu cầu của dự án với chi phí tối thiểu. Chi phí bao gồm chi phí quản lý sợi thép và hao hụt do các đoạn phế liệu.

Ta có thể phát biểu bài toán này như bài toán tìm đường đi ngắn nhất như sau. Ta xây dựng một đồ thị có hướng gồm (n+1) nút đánh số từ $0 \rightarrow n$. Mỗi nút trong đồ thị tương ứng với các loại sợi thép có chiều dài khác nhau. Nút 0 tương ứng với sợi thép có chiều dài 0 (zero) và nút n tương ứng với sợi thép dài nhất. Với mỗi nút i, ta dựng các cung nối i với mọi nút j = i+1, i+2, ..., n. Cung (i,j) cho biết rằng ta sẽ lưu trữ các sợi thép với chiều dài L_j trong kho và dùng nó để đáp ứng yêu cầu của dự án về các sợi thép loại có chiều dài $L_{i+1}, L_{i+2}, ..., L_j$. Chi phí (trọng) c_{ij} của cung (i,j) là

$$c_{ij} = K_j + \sum_{k=i+1}^{j} D_k$$

Chi phí của cung (i,j) gồm 2 phần: (1) chi phí cố định K_j của việc lưu trữ thép loại chiều dài L_j, và (2) chi phí sử dụng sợi thép loại L_j để đáp ứng nhu cầu về sợi thép loại $L_{i+1}, L_{i+2}, ..., L_j$. Một đường đi từ nút 0 đến nút n cho biết một phương thức đáp ứng yêu cầu của dự án. Đường đi ngắn nhất từ 0 đến n cho biết phương án với chi phí tối thiểu cần tìm.

3.3. Ứng dụng 3: *Bài toán lịch thi đấu*

Xem xét một giải thi đấu giữa **n** đội thể thao. Giả sử rằng, hai đội bất kỳ thi đấu với nhau đúng **c** trận và các trận đấu không hòa (luôn phân định thắng thua). Một người nói rằng, sau khi hoàn tất giải, đội i thắng đúng α_i trận ($1 \le i \le n$). Ta cần kiểm chứng xem liệu bộ số $\alpha_1, \alpha_2, ..., \alpha_i, ..., \alpha_n$ có khả năng biểu diễn kết quả thi đấu của một giải như vậy không ?

Ta định nghĩa một đồ thị có hướng G = (N, A) với tập đỉnh N = {1,2, ..., n} biểu diễn cho n đội và tập cung A = {(i,j) ∈ NxN : i < j}. Như vậy, mỗi nút i được nối với các nút i+1, i+2, ..., n. Gọi x_{ij} là số lần đội i thắng đội j (i < j). Như vậy, số lần đội i thắng các đội i+1, i+2, ..., n là $\sum_{\{j:(i,j)\in A\}} x_{ij}$. Ngoài ra, số lần đội i thắng đội j (i>j) sẽ là c-x_{ij}. Từ đây ta có, số lần đội i thắng các đội 1, 2, ..., i-1 là $(1-i)c - \sum_{\{j:(j,i)\in A\}} x_{ji}$. Tổng số trận thắng của đội i phải bằng tổng các trận thắng các đội 1, 2, ..., n. Từ nhận xét này ta có phương trình:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \alpha_{ij} - (1-i)c \quad \text{với mọi } i \in N. \qquad (5.1)$$

Hơn nữa, các giá trị x_{ij} còn phải thỏa mãn ràng buộc:

$$0 \le x_{ij} \le c \qquad \text{với mọi } (i, j) \in A \qquad (5.2)$$

Những phân tích trên cho ta thấy rằng, các giá trị α_i chỉ có thể là kết quả thi đấu của giải khi các ràng buộc (5.1), (5.2) có ít nhất một lời giải x thỏa. Đặt b(i) = α_i-(i-1)c. Lưu ý rằng ta có đẳng thức sau:

$$\sum_{i\in N} \alpha_i = \sum_{i\in N} (i-1)c = \frac{cn(n-1)}{2}$$

như vậy ta có thể dễ dàng suy ra:

$$\sum_{i\in N} b(i) = 0$$

Với cách phát biểu này, bài toán tìm một nghiệm tương thích với hệ dạng (5.1) & (5.2) được gọi là *bài toán luồng tương thích* và có thể giải bằng cách giải một bài toán luồng cực đại (xin xem ở phần cuối chương).

3.4. Ứng dụng 4: *Bài toán làm tròn số trên ma trận*

Bài toán này liên quan đến vấn đề làm tròn giá trị của các phần tử, tổng giá trị các phần tử trên dòng, tổng giá trị các phần tử trên cột của một ma trận. Ta có một ma trận kích thước p×q các số thực D = (d_{ij}) với các giá trị tổng dòng, tổng cột lần lượt là α_i và β_j. Ta có thể làm tròn một số thực bất kỳ a thành số nguyên lớn nhất nhỏ hơn hay bằng nó $\lfloor a \rfloor$ (phần nguyên), hoặc thành số nguyên nhỏ nhất lớn hơn hay bằng nó $\lceil a \rceil$ (số trần), và việc quyết định làm tròn lên hay xuống hoàn toàn phụ thuộc vào quyết định của chúng ta. Bài toán làm tròn ma trận đòi hỏi chúng ta phải làm tròn các phẩn tử của ma trận và các giá trị tổng dòng, tổng cột sao cho tổng các phần tử trên mỗi dòng cũng như mỗi cột của ma trận đã được làm tròn bằng với giá trị làm tròn của các tổng dòng, tổng cột tương ứng. Phép làm tròn như vậy ta gọi là phép làm tròn ma trận.

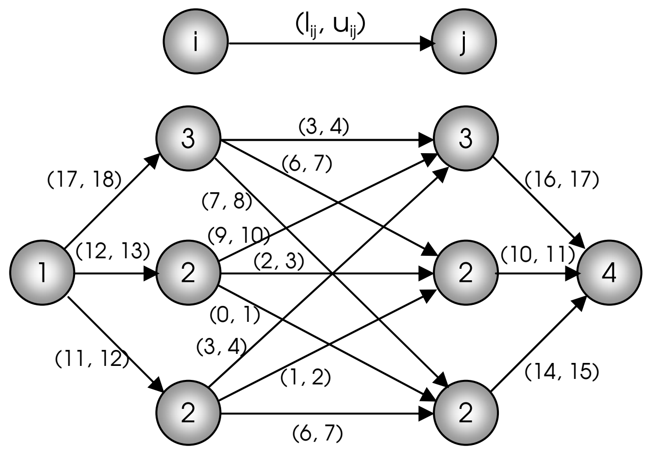
Ta sẽ chỉ ra cách giải bài toán này bằng cách đưa về việc giải một bài toán luồng có chặn dưới (luồng trên cung với khả năng thông qua tối thiểu lớn hơn không). Và trong phần sau, các bạn có thể thấy rằng, bài toán luồng có chặn dưới sẽ được giải bằng hai bài toán luồng cực đại bình thường.

Để minh họa cho phương pháp giải, ta xét ví dụ trong hình 5.1 dưới đây:

				Tổng dòng
	3.1	6.8	7.3	**17.2**
	9.6	2.4	0.7	**12.7**
	3.6	1.2	6.5	**11.3**
Tổng cột:	**16.3**	**10.4**	**14.5**	

Hình 5.1 - Bài toán làm tròn ma trận

Hình 5.2 biểu diễn mạng luồng cực đại tương ứng với bài toán trên. Mạng này chứa các nút i tương ứng với các dòng và các nút j’ tương ứng với các cột. Cung (i, j’) trong mạng tương ứng với mỗi phần tử d_{ij} của ma trận và các cung (s, i) tương ứng với các tổng dòng còn các cung (j’, t) tương ứng với các tổng cột. Khả năng thông qua tối thiểu và tối đa của cung (i,j’) tương ứng là $\lfloor d_{ij} \rfloor$ và $\lceil d_{ij} \rceil$. Dễ dàng nhận thấy sự tương ứng 1-1 giữa lời giải của bài toán và một luồng tương thích của mạng trong hình 5.2. Như vậy, ta có thể tìm lời giải của bài toán làm tròn ma trận bằng cách giải bài toán luồng cực đại trên mạng tương ứng.



Hình 5.2 - Mạng cho bài toán làm tròn ma trận

Bài toán làm tròn ma trận này xuất hiện trong một số ứng dụng thực tế. Ví dụ, Ủy ban điều tra dân số có thể sử dụng các thông tin điều tra được để lập thành hàng triệu bảng dùng trong các mục đích khác nhau. Theo luật định, Ủy ban có nghĩa vụ bảo mật các nguồn cung cấp thông tin và không được công bố các thông tin mang tính riêng tư của bất kỳ cá nhân nào. Chúng ta có thể che dấu các thông tin như sau. Ta sẽ làm tròn các số liệu trong bảng, kể các các tổng dòng, tổng cột thành một bội số của hằng số nguyên dương k định trước nào đó tương tự như bài toán làm tròn ma trận ở trên. Chỉ khác là, thay vì làm tròn thành các bội số của 1 thì bây giờ ta làm tròn thành bội số của một hằng số $k \geq 1$. Nếu biết giá trị của k, ta có thể giải quyết bài toán bắng cách xây dựng mạng tương ứng giống như đã làm, nhưng bây giờ, khả năng thông qua tối thiểu và tối đa của cung (i,j') tương ứng là bội số lớn nhất nhỏ hơn hay bằng d_{ij} và bội số nhỏ nhất lớn hơn hay bằng d_{ij}.

3.5. Ứng dụng 5: phân bố tính toán trên máy tính có hai bộ vi xử lý

Ứng dụng này xem xét sự phân bố việc thực hiện các module khác nhau của một chương trình trên máy tính có 2 bộ vi xử lý sao cho tổng chi phí thực hiện chương trình là tối thiểu. Chi phí thực hiện sẽ bao gồm chi phí tính toán trên các module và chi phí truyền thông giữa các module. Chi phí thực hiện một module trên 2 bộ vi xử lý sẽ khác nhau bởi bộ nhớ đi kèm, tốc độ tính toán, khả năng xử lý số học, ... Giả sử α_i và β_i ký hiệu chi phí tính toán của module i lần lượt trên bộ vi xử lý 1 và 2. Khi 2 module i, j được thực hiện trên 2 bộ vi xử lý khác nhau ta sẽ tốn thêm chi phí truyền thông giữa 2 module c_{ij}. Nhiệm vụ của chúng ta là chọn lựa nơi thực hiện các module sao cho tổng chi phí là tối thiểu.

i	1	2	3	4
α_i	6	5	10	4
β_i	4	10	3	8

(a)

$$(c_{ij}) = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 0 & 0 \\ 2 & 5 & 0 & 6 & 2 \\ 3 & 0 & 6 & 0 & 1 \\ 4 & 0 & 2 & 1 & 0 \end{array}$$

(b)

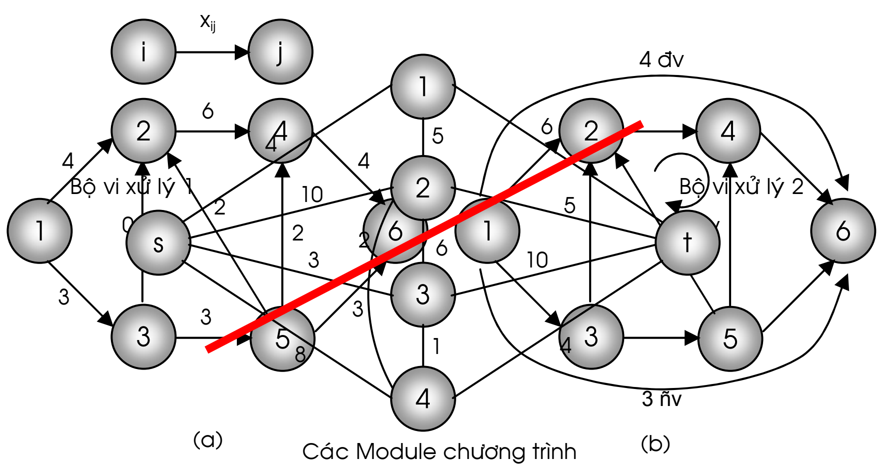
Hình 5.3 Dữ liệu cho mô hình tính toán phân bố trên 2 bộ vi xử lý

Ta sẽ biểu diễn bài toán này như là bài toán lát cắt nhỏ nhất trên mạng vô hướng như sau. Ta định nghĩa nút nguồn s tương ứng với bộ vi xử lý 1, nút đích t tương ứng với bộ vi xử lý 2 và mỗi module sẽ có một nút tương ứng trong mạng. Với mỗi nút i khác s và t, ta nối cung (s,i) với khả năng thông qua là β_i và cung (i,t) với khả năng thông qua là α_i. Cuối cùng, nếu module i có giao tiếp với module j, ta thêm cung (i,j) với khả năng thông qua là c_{ij}. Hình 5.3 và hình 5.4 cho ta một ví dụ về bài toán trên. Hình 5.3 cho biết dữ liệu của bài toán và hình 5.4 minh họa mạng tương ứng.

Với cách biểu diễn này, ta có sự tương ứng 1-1 giữa một lát cắt s-t và một cách phân công việc thực hiện các module trên 2 bộ vi xử lý. Giá trị của lát cắt chính là chi phí của cách phân công tương ứng (xem hình vẽ 5.4).

4. PHÂN RÃ LUỒNG

Khi phát biểu bài toán luồng trên mạng, ta có thể sử dụng 2 hướng tiếp cận tương đương: (1) ta có thể định nghĩa luồng thông qua giá trị luồng trên các cung (arc flow) của mạng (như đề cập trong phần II) hoặc (2) ta có thể



Hình 5.4 – Mạng cho mô hình phân bố
Hình 5.5 – Hai cách mô tả luồng

định nghĩa luồng thông qua giá trị luồng trên các đường đi và chu trình (path and circle flow). Ví dụ, luồng trên mạng trong hình 5.5a định nghĩa theo hướng tiếp cận (1) gửi 7 đơn vị luồng từ nút 1 đến nút 6. Luồng trên mạng trong hình 5.5b định nghĩa theo hướng tiếp cận (2) tương ứng với luồng trong hình 5.5a. Trong 5.5b, ta gửi 4 đơn vị luồng dọc theo đường đi 1-2-4-6, 2 đơn vị luồng dọc theo đường đi 1-3-5-6, và 2 đơn vị luồng dọc theo chu trình 2-4-5-2. Mặc dù trong hầu hết giáo trình này, chúng ta sử dụng dạng luồng trên cung, nhưng trong một số trường hợp, ta sẽ cần biểu diễn dạng đường đi và chu trình của luồng hoặc các kết quả nảy sinh từ hướng tiếp cận này. Trong phần này, chúng ta sẽ xem xét một số mối liên hệ giữa 2 hướng tiếp cận này.

Trong phần này, một luồng trên cung chúng ta hiểu là một vectơ x = {x_{ij} } thỏa các ràng buộc sau :

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = -e(i) \quad \forall i \in N \qquad (5.3a)$$

$$0 \le x_{ij} \le u_{ij} \text{ với mọi } (i, j) \in A \qquad (5.3b)$$

trong đó $\sum_{x=1}^{n} e(i) = 0$. Chú ý rằng trong mô hình này chỉ số cung và cầu b(i) của nút i chính là –e(i); chúng ta xem e(i) như là lượng mất cân đối của nút i. Chúng ta chọn một mô hình khác thay thế bởi vì một số thuật giải luồng cực đại và cực tiểu được mô tả trong sách này đưa ra các lời giải thỏa các ràng buộc của luồng nhưng không nhất thiết thỏa các ràng buộc về cung và cầu. Giá trị e(i) là hiệu các giá trị luồng vào và luồng ra tại nút i. Nếu giá trị của luồng vào lớn hơn giá trị của luồng ra tại nút i thì e(i) > 0 và chúng ta nói rằng nút i là nút *thừa*. Ngược lại, nếu giá trị của luồng vào nhỏ hơn giá trị của luồng ra thì e(i) < 0 và chúng ta nói rằng nút i là nút *thiếu*. Nếu giá trị luồng vào và giá trị luồng ra bằng nhau thì ta gọi nút i là nút *cân bằng*. Chú ý rằng, nếu e = -b thì luồng x là một luồng tương thích của bài toán luồng cực tiểu.

Trong phương pháp biểu diễn luồng không qua giá trị luồng trên cung, các biến cơ sở là các giá trị luồng x_{ij} trên cung (i, j) ∈ A. Biểu diễn luồng thông qua đường đi và chu trình bắt đầu với một danh sách tất cả các đường đi có hướng p giữa hai cặp nút bất kỳ và tất cả các chu trình có hướng w của mạng. Gọi P là tập hợp tất cả các đường đi và W là tập hợp tất cả các chu trình. Các biến cơ sở trong biểu diễn luồng thông qua đường đi và chu trình là : f(p) - giá trị luồng trên đường đi p, f(w) - giá trị luồng trên chu trình w; chúng ta định nghĩa các biến này cho mỗi đường đi có hướng p trong P và mỗi chu trình có hướng w trong W.

Chú ý rằng với mỗi tập luồng trên đường đi và chu trình định nghĩa một cách duy nhất các giá trị luồng trên cung một cách tự nhiên : giá trị luồng x_{ij} trên cung (i, j) bằng với tổng của các giá trị luồng f(p) và f(w) của tất cả các đường đi p và các chu trình w có chứa cung này. Chúng ta hình thức hóa nhận xét trên bằng cách định nghĩa một số ký hiệu mới: $\delta_{ij}(p) = 1$ nếu cung $(i, j) \in p$, và ngược lại sẽ bằng 0. Tương tự như vậy, $\delta_{ij}(w) = 1$ nếu cung $(i, j) \in W$, và bằng 0 trong trường hợp ngược lại. Khi đó,

$$x_{ij} = \sum_{p \in P} \delta_{ij}(p) f(p) + \sum_{w \in W} \delta_{ij}(w) f(w)$$

Do đó, mỗi luồng trên đường đi và chu trình xác định duy nhất các giá trị luồng trên cung. Chúng ta có thể quay ngược qui trình này được không? Nghĩa là, chúng ta có thể phân rã bất kỳ một luồng trên cung nào thành luồng trên đường đi và chu trình không? Các định lý sau sẽ đưa ra giải đáp cho câu hỏi này.

Định lý 5.1: (Định lý về sự phân rã luồng)

Mỗi luồng trên đường đi và chu trình được biểu diễn duy nhất bằng các giá trị luồng không âm trên cung. Ngược lại, mỗi luồng không âm trên cung x có thể được biểu diễn như là một luồng đường đi và chu trình (không duy nhất) với hai tính chất sau :

a) Mỗi đường đi có hướng với giá trị luồng dương nối một nút *thiếu* với một nút *thừa*.

b) Có nhiều nhất n + m đường đi và chu trình có giá trị luồng khác 0; hơn nữa, có tối đa m chu trình có giá trị luồng khác 0.

Chứng minh: trước tiên, chúng ta nhận thấy rằng chỉ cần phải chứng minh phản chứng. Chúng ta sẽ đưa ra một chứng minh dạng thuật toán để chỉ ra cách phân rã một luồng trên cung x bất kỳ thành một luồng trên đường đi và chu trình. Giả sử rằng i_0 là nút *thiếu*. Khi đó một cung (i_0, i_1) nào đó mang giá trị luồng dương. Nếu i_1 là một nút *thừa* thì thuật toán ngừng; ngược lại, ràng buộc (5.3a) của nút i_1 bảo đảm rằng tồn tại một cung (i_1, i_2) khác mang giá trị luồng dương. Chúng ta lặp lại thao tác trên cho đến khi gặp phải một nút *thừa* hoặc là chúng ta sẽ gặp lại nút đã đi qua rồi. Chú ý rằng một trong hai trường hợp này sẽ xảy ra sau tối đa sau n lần thực hiện thao tác trên. Trong trường hợp 1, chúng ta có một đường đi có hướng p từ một nút *thiếu* i_0 tới nút *thừa* i_k và trong trường hợp 2, chúng ta sẽ có một chu trình có hướng w. Trong bất kỳ trường hợp nào, đường đi hoặc chu trình nhận được sẽ chứa toàn các cung có giá trị luồng dương. Nếu chúng ta nhận được một đường đi có hướng, ta có $f(p) = \min\{-e(i_0), e(i_k), \min\{x_{ij} : (i, j) \in P\}\}$ và cập nhật lại $e(i_0) = e(i_0) + f(p)$, $e(i_k) = e(i_k) - f(p)$, và $x_{ij} = x_{ij} - f(p)$ cho mỗi cung (i, j) trong p. Nếu chúng ta nhận được một chu trình w, ta có $f(w) = \min\{x_{ij} : (i, j) \in w\}$ và cập nhật lại $x_{ij} = x_{ij} - f(w)$ trong mỗi cung (i, j) trong w.

Chúng ta lặp lại quá trình trên cho bài toán đã được cập nhật lại cho đến khi tất cả các nút đều cân bằng. Khi đó, chúng ta chọn một nút bất kỳ có ít nhất một cung ra với giá trị luồng dương như là nút bắt đầu và lặp lại quy trình trên. Lúc này, chúng ta phải tìm được một chu trình có hướng. Chúng ta sẽ ngừng lại khi x = 0. Rõ ràng, luồng nguyên thủy bằng tổng của các luồng trên đường đi và chu trình xác định được bởi phương pháp này. Bây giờ, hãy chú ý rằng, mỗi khi chúng ta xác định được một đường đi có hướng, chúng ta sẽ biến một nút *thừa* hay *thiếu* thành nút *cân bằng* hoặc giảm giá trị luồng của một cung nào đó về 0; mỗi khi chúng ta xác định được một chu trình có hướng, chúng ta sẽ giảm giá trị luồng của một cung nào đó về 0. Kết quả là, luồng trên đường đi và chu trình luồng x thì chứa tổng cộng nhiều nhất n + m đường đi có hướng và chu trình, và nhiều nhất m trong số này là chu trình có hướng.

Chúng ta hãy xem xét một luồng x mà $e(i) = 0\ \forall i \in N$. Nhắc lại rằng trong phần II chúng ta bài toán này gọi là bài toán luồng lưu thông. Khi chúng ta áp dụng thuật giải phân rã luồng đối với bài toán luồng lưu thông, với mỗi lần lặp ta sẽ phát hiện ra một chu trình có hướng chứa toàn là các cung có giá trị luồng dương và hệ quả là luồng trên ít nhất một cung giảm xuống 0. Kết quả là, một luồng lưu thông sẽ được phân rã tối đa thành m giá trị luồng trên chu trình có hướng.

5. LUỒNG CỰC ĐẠI

Một trong những bài toán luồng cực tiểu có nhiều ứng dụng nhất là bài toán luồng cực đại. Luồng cực đại được phát biểu như sau: Trong một mạng với các cung có độ thông qua u_{ij}, ta muốn gửi một một luồng có giá trị cực đại giữa 2 nút đặc biệt, nút nguồn s và nút đích t mà không vượt quá khả năng thông qua của các cung.

5.1. Các ký hiệu và giả thiết

Ta sẽ xem xét một mạng G=(N,A) với các độ thông qua không âm u_{ij} của các cung $(i,j) \in A$. Giả sử $U = \max\{u_{ij}: (i,j) \in A\}$. Ta ký hiệu $A(i) = \{(i,k): (i,k) \in A\}$ là tập hợp các cung đi ra từ i. Để định nghĩa bài toán luồng cực đại, ta phân biệt 2 nút đặc biệt trong mạng G: *nút nguồn s* và *nút đích t*. Ta muốn tìm luồng cực đại gửi từ nút nguồn s đến nút đích t thỏa mãn mọi ràng buộc về khả năng thông qua của các cung. Ta có thể biểu diễn bài toán một cách hình thức như sau:

Cực đại hóa giá trị luồng v (5.4a)

thỏa mãn ràng buộc:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} v & i = s, \\ 0 & \forall i \in N - \{s,t\} \\ -v & i = t \end{cases} \tag{5.4b}$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in A. \tag{5.4c}$$

Ta gọi vector $x = \{x_{ij}\}$ thỏa mãn (5.4b) và (5.4c) là luồng và giá trị v tương ứng gọi là giá trị của luồng. Ta sẽ xem xét bài toán luồng cực đại với các giả thiết sau:

- *Mạng G là mạng có hướng.*
- *Tất cả các độ thông qua của các cung trong mạng đều là các số nguyên không âm.* Giả thiết này hoàn toàn không làm mất tính tổng quát của bài toán do các máy tính hiện tại để biểu diễn các số thực bằng các số hữu tỉ.
- *Mạng không chứa một đường đi từ s đến t mà trong đó tất cả các cung đều có độ thông qua là ∞.*
- *Nếu cung (i,j) thuộc A thì cung (j,i) cũng thuộc A.* Giả thiết này

không bắt buộc vì trong mạng có thể chứa các cung với độ thông qua uij = 0.

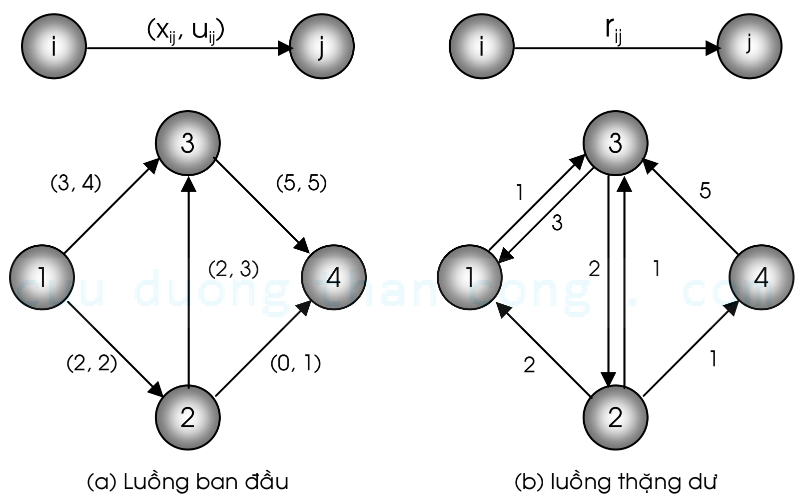
- *Mạng không chứa các cung song song*. Giả thiết này chỉ đơn thuần vì sự tiện lợi trong việc xét bài toán. Nó không làm mất tính tổng quát của bài toán (tại sao ?).

5.2. Luồng và lát cắt (flows and cuts)

Trong phần này, chúng ta sẽ xem xét một số thuộc tính cơ bản của luồng và lát cắt. Ta sẽ dùng các thuộc tính này để chứng minh định lý quan trọng nhất về luồng cực đại, định lý về lát cắt tối thiểu (max-flow min-cut theorem).

5.2.1. Mạng thặng dư (residual network)

Khái niệm mạng thặng dư sẽ đóng vai trò trung tâm trong quá trình xây dựng các thuật toán tìm luồng cực đại mà chúng ta sẽ xem xét trong giáo trình này.

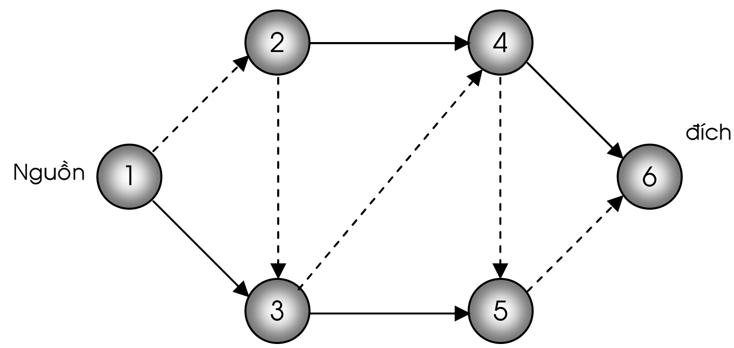


Hình 5.6 Minh họa luồng thặng dư

Giả sử có một luồng x, khả năng thông qua thặng dư (residual capacity) r_{ij} của một cung (i,j) bất kỳ là giá trị luồng tối đa có thể gửi từ nút i đến nút j thông qua cung (i,j) và cung (j,i). Khả năng thông qua thặng dư r_{ij} có 2 thành phần: (1) $u_{ij} - x_{ij}$ là khả năng thông qua còn chưa dùng của cung (i,j), và (2) giá trị luồng hiện hành x_{ji} trên cung (j,i) và ta có thể loại bỏ (cancel) để tăng luồng từ nút i đến nút j. Như vậy, $r_{ij} = u_{ij} - x_{ij} + x_{ji}$. Ta ký hiệu G(x) là mạng chứa các cung với độ thông qua thặng dư r_{ij} ứng với luồng x. Ta gọi G(x) là mạng thặng dư. Hình 5.6 cho một ví dụ về mạng thặng dư.

5.2.2. Lát cắt s-t

Lát cắt là một phân hoạch chia tập nút N thành 2 tập con S và $\overline{S} = N-S$. Ta ký hiệu lát cắt này là $[S, \overline{S}]$. Ta cũng có thể định nghĩa lát cắt là một



Hình 5.7 Ví dụ một lát cắt s-t

tập hợp các cung mà các nút kề với chúng thuộc 2 tập nút con S và $\overline{S}$. ta gọi một lát cắt là lát cắt s-t nếu $s \in S$ và $t \in \overline{S}$. Ta cũng gọi cung (i,j) với $i \in S$ và $j \in S'$ là cung tới và gọi cung (i,j) với $i \in \overline{S}$ và $j \in S$ là cung lùi của lát cắt $[S, \overline{S}]$. Giả sử $(S, \overline{S})$ là tập hợp các cung tới trong lát cắt và $(\overline{S}, S)$ là tập hợp các cung lùi. Ví dụ, trong hình 5.7, các cung biểu diễn bằng đường gạch nối xác định một lát cắt s-t. Với lát cắt này, $(S, \overline{S}) = \{(1,2),(3,4),(5,6)\}$ và $(\overline{S}, S) = \{(2,3), (4,5)\}$.

Ta định nghĩa *độ thông qua của một lát cắt s-t* [S, $\bar{S}$], ký hiệu là u[S, $\bar{S}$], là tổng của các độ thông qua của các cung tới trong lát cắt. Nghĩa là

$$u\left[S,\bar{S}\right]=\sum_{(i,j)\in(S,\bar{S})} u_{ij}$$

Rõ ràng, độ thông qua của một lát cắt s-t là chặn trên của giá trị luồng tối đa ta có thể gửi từ các nút trong S đến các nút trong $\bar{S}$.

Lát cắt tối thiểu là lát cắt s-t có độ thông qua nhỏ nhất.

Ta định nghĩa *độ thông qua thặng dư của một lát cắt s-t* [S,S'], ký hiệu là r[S, $\bar{S}$], là tổng của các độ thông qua thặng dư của các cung tới trong lát cắt. Nghĩa là

$$r\left[S,\bar{S}\right]=\sum_{(i,j)\in(S,\bar{S})} r_{ij}$$

Luồng băng qua một lát cắt s-t: giả sử x là một luồng trong mạng G. Cộng 2 vế của đẳng thức (5.4b) ứng với các nút trong S, ta nhận được:

$$v=\sum_{i\in S}\left[\sum_{\{j:(i,j)\in A\}} x_{ij}-\sum_{\{j:(j,i)\in A\}} x_{ji}\right]$$

Ta có thể đơn giản hóa biểu thức trên nhờ lưu ý đến tính chất sau: nếu nút p và nút q cùng thuộc tập S và (p.q) ∈ A, biến x_{pq} trong tổng đầu tiên sẽ triệt tiêu biến $-x_{qp}$ trong tổng thứ hai. Hơn thế nữa, nếu cả 2 nút p, q cùng thuộc $\bar{S}$, thì x_{pq} sẽ không xuất hiện trong biểu thức. Như vậy:

$$v=\sum_{(i,j)\in(S,\bar{S})} x_{ij}-\sum_{(i,j)\in(\bar{S},S)} x_{ij} \qquad (5.5)$$

Tổng đầu tiên trong vế phải của (5.5) biểu diễn lượng luồng truyền từ các nút trong S đến các nút trong $\bar{S}$ và tổng thứ 2 biểu diễn lượng luồng truyền từ các nút trong $\bar{S}$ đến các nút trong S. Như vậy, vế phải của (5.5) biểu diễn tổng giá trị luồng băng qua lát cắt, và (5.5) hàm ý rằng luồng băng qua một lát cắt s-t bất kỳ bằng v. Do $0 \le x_{ij} \le u_{ij}$ nên từ (5.5) suy ra:

$$v \le \sum_{(i,j)\in(S,\bar{S})} u_{ij} = u[S,\bar{S}] \tag{5.6}$$

Bất đẳng thức (5.6) cho phép ta khẳng định rằng giá trị của một luồng bất kỳ nhỏ hơn hay bằng khả năng thông qua của một lát cắt s-t bất kỳ trong mạng.

Mệnh đề 5.1: *Giá trị của một luồng bất kỳ nhỏ hơn hay bằng khả năng thông qua của mọi lát cắt trong mạng.*

Mệnh đề trên hàm ý rằng nếu ta phát hiện được một luồng x mà giá trị của nó bằng khả năng thông qua của một lát cắt [S, $\bar{S}$] nào đó, thì x chính là luồng cực đại và lát cắt [S, $\bar{S}$] chính là lát cắt tối thiểu. Định lý về lát cắt tối thiểu, sẽ được chứng minh trong phần kế, khẳng định rằng, một luồng bất kỳ luôn cos giá trị luồng bằng khả năng thông qua của một lát cắt nào đó.

Tiếp theo, chúng ta sẽ phát biểu lại mệnh đề 5.1 theo thuật ngữ của độ thông qua thặng dư. Giả sử rằng x là một luồng có giá trị v và x' là một luồng có giá trị v + Δv (Δv > 0). Bất đẳng thức (5.6) cho ta:

$$v + \Delta v \le \sum_{(i,j)\in(S,\bar{S})} u_{ij} \tag{5.7}$$

Lấy (5.7) trừ (5.5) ta được:

$$\Delta v \le \sum_{(i,j)\in(S,\bar{S})} (u_{ij} - x_{ij}) + \sum_{(i,j)\in(\bar{S},S)} x_{ij} \tag{5.8}$$

Bây giờ, ta sử dụng giả thiết 5.4 để có thể viết $\sum_{(i,j)\in(\bar{S},S)} x_{ij}$ thành $\sum_{(i,j)\in(S,\bar{S})} x_{ji}$. Như vậy:

$$\Delta v \le \sum_{(i,j)\in(S,\bar{S})} (u_{ij} - x_{ij} + x_{ji}) = \sum_{(i,j)\in(S,\bar{S})} r_{ij} \tag{5.9}$$

Mệnh đề sau đây trở nên hiển nhiên:

Mệnh đề 5.2: *Với mọi luồng x có giá trị v trong mạng, lượng luồng có thể gửi thêm từ s đến t luôn nhỏ hơn khả năng thông qua thặng dư của một lát cắt s-t bất kỳ.*

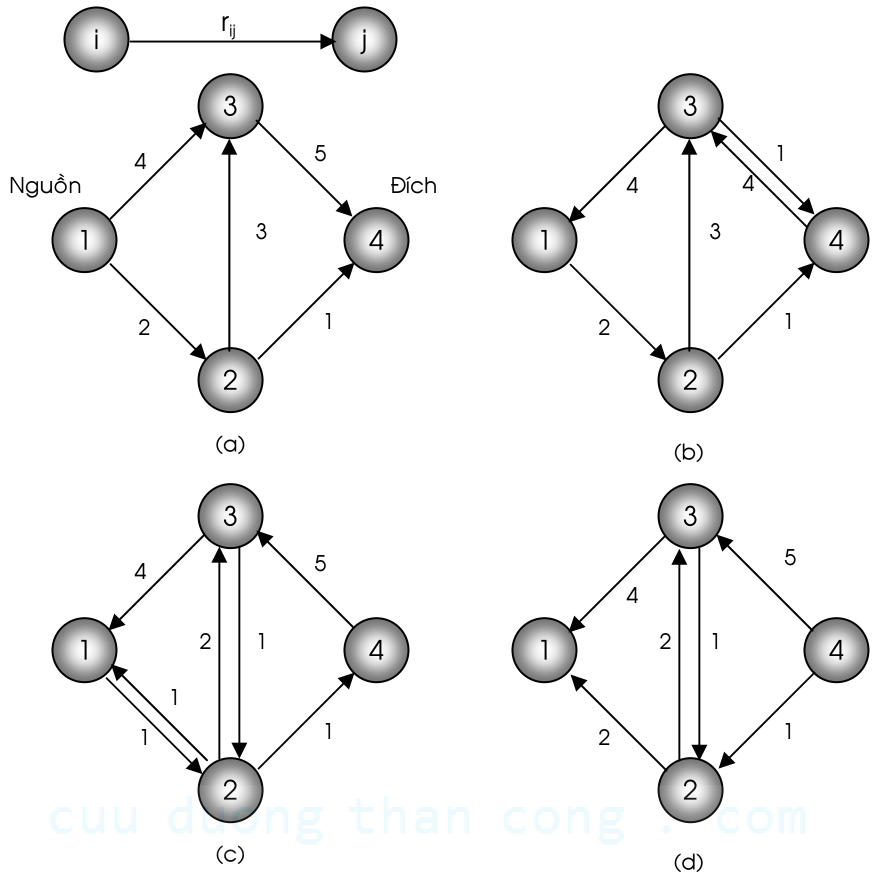
5.3. Thuật toán đường tăng trưởng (augmenting path) tìm luồng cực đại

Trong mục này, chúng ta sẽ tìm hiểu một thuật toán sơ khai nhất để giải bài toán luồng cực đại. Thuật toán này được biết với tên gọi thuật toán đường tăng trưởng (augmenting path algorithm).

Ta gọi một đường đi có hướng từ s đến t trong mạng thặng dư là một đường tăng trưởng (augmenting path). Ta gọi độ thông qua thặng dư của một đường tăng trưởng là độ thông qua thặng dư nhỏ nhất của các cung trong con đường này. Ví dụ, trong mạng thặng dư trình bày ở hình 5.5b chứa duy nhất 1 đường tăng trưởng 1-3-2-4 và độ thông qua thặng dư của nó $\delta = \min\{r_{13}, r_{32}, r_{24}\} = \min\{1,2,1\} = 1$. Như vậy độ thông qua thặng dư δ của một đường tăng trưởng luôn dương. Vì vậy, nếu mạng thặng dư chứa một đường tăng trưởng, ta có thể tăng giá trị luồng gửi từ s đến t trong mạng. Thuật toán đường tăng trưởng dựa trên cơ sở của nhận xét vừa nêu. Thuật toán hoạt động theo qui trình lập đi lập lại việc tìm một đường tăng trưởng và tăng giá trị luồng gửi từ s đến t dọc theo con đường này.

```
Thuật toán Augmenting path
Begin
   x := 0;
   while (G(x) chứa một đường đi có hướng từ s đến t) do
   begin
       Xác định đường tăng trưởng P từ s đến t;
       δ := min{r_ij: (i,j) ∈ P};
       Tăng luồng thêm giá trị là δ dọc theo P và cập nhật G(x);
```

```
    end;
End;
```



Hình 5.8 - Minh họa thuật toán đường tăng trưởng

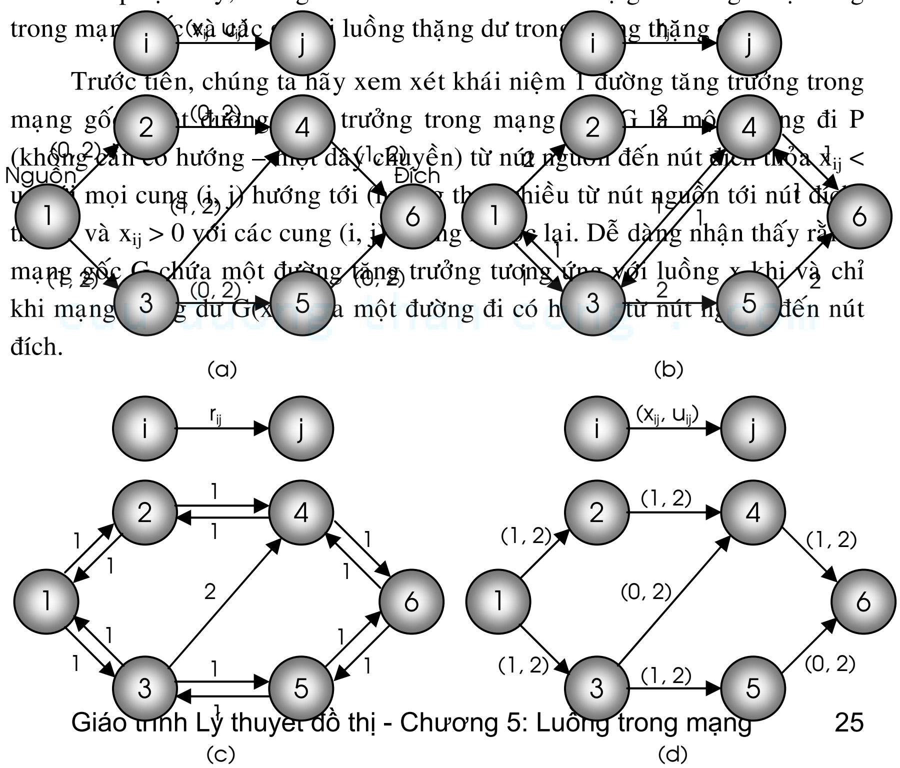
Chúng ta sẽ dùng mạng trong hình 5.8a để minh họa việc giải bài toán luồng cực đại bằng thuật toán đường tăng trưởng. Giả sử rằng thuật toán chọn đường đi 1-3-4 để tăng trưởng. Độ thông qua thặng dư của con đường này là $\delta = \min\{r_{13}, r_{34}\} = \min\{4, 5\} = 4$. Bước tăng trưởng này giảm độ thông qua thặng dư cuả cung (1,3) đi 4 đơn vị.

Độ thông qua thặng dư của cung (3,4) cũng giảm từ 5 xuống còn 1 và độ thông qua thặng dư của cung (4,3) tăng từ 0 lên 4. Hình 5.8b minh họa mạng thặng dư ở giai đoạn này. Ở lần lặp thứ 2, giả sử rằng thuật toán chọn đường đi 1-2-3-4. Độ thông qua thặng dư của con đường này $\delta = \min\{2, 3, 1\} = 1$. Tăng trưởng 1 đơn vị luồng dọc theo con đường này, trạng thái của luồng thặng dư sẽ giống như trong hình 5.8c. Trong lần lặp thứ 3, thuật toán tăng trưởng 1 đơn vị luồng dọc theo đường 1-2-4. Hình 5.8d chỉ ra luồng thặng dư tương ứng. Lúc này, luồng thặng dư không còn chứa con đường nào từ 1 đến 4, nên thuật toán dừng ở đây.

5.3.1. Mối quan hệ giữa mạng gốc và mạng thặng dư

Trong khi cài đặt thuật toán đường tăng trưởng, ta có thể chọn lựa làm việc trực tiếp trên mạng gốc với luồng x_{ij} hoặc xây dựng mạng thặng dư G(x) và theo dõi sự thay đổi của các giá trị luồng thặng dư r_{ij}. Và sau đó, khi thuật toán dừng, ta sẽ khôi phục giá trị luồng x_{ij} dựa vào r_{ij}. Để nắm vững cả 2 cách tiếp cận này, chúng ta cần tìm hiểu mối liên hệ giữa các giá trị luồng trong mạng gốc và các giá trị luồng thặng dư trong mạng thặng dư.

Trước tiên, chúng ta hãy xem xét khái niệm 1 đường tăng trưởng trong mạng gốc. Một đường tăng trưởng trong mạng gốc G là một đường đi P (không cần có hướng – một dây chuyền) từ nút nguồn đến nút đích thỏa $x_{ij} < u_{ij}$ với mọi cung (i, j) hướng tới (hướng theo chiều từ nút nguồn tới nút đích) và $x_{ij} > 0$ với các cung (i, j) hướng ngược lại. Dễ dàng nhận thấy rằng mạng gốc G chứa một đường tăng trưởng tương ứng với luồng x khi và chỉ khi mạng thặng dư G(x) chứa một đường đi có hướng từ nút nguồn đến nút đích.



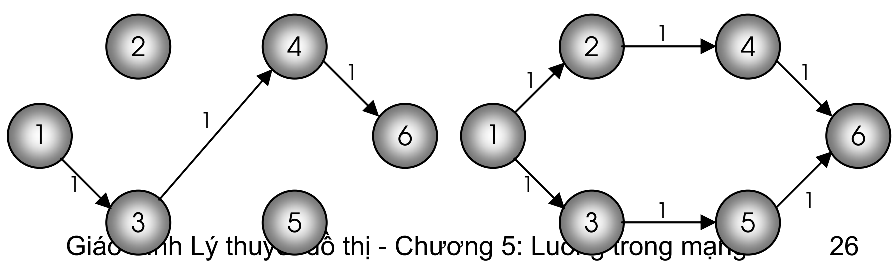
Hình 5.9 Ảnh hưởng của phép tăng trưởng lên sự phân rã luồng

Bây giờ, giả sử rằng ta cập nhật các giá trị luồng thặng dư tại một thời điểm nào đó trong quá trình thực thi thuật toán. Lúc đó x_{ij} sẽ bị ảnh hưởng ra sao ? Định nghĩa của luồng thặng dư ($r_{ij} = u_{ij} - x_{ij} + x_{ji}$) xác nhận rằng, việc tăng δ đơn vị luồng trên cung (i, j) trong mạng thặng dư tương ứng với (1) việc tăng x_{ij} thêm δ đơn vị luồng hoặc (2) giảm x_{ji} đi δ đơn vị luồng hay (3) một tổ hợp tuyến tính giữa (1) và (2) nghĩa là x_{ij} tăng thêm δ_1 đơn vị còn x_{ji} giảm đi δ_2 đơn vị với $\delta_1 + \delta_2 = \delta$. Chúng ta sẽ dùng ví dụ trong hình 5.9 để minh họa những trường hợp này. Việc tăng trưởng 1 đơn vị luồng dọc theo đường 1-2-4-3-5-6 trong mạng sinh ra mạng thặng dư trong hình 5.9c với các giá trị luồng x_{ij} tương ứng như trong hình 5.9d. So sánh lời giải trong hình 5.9d với mạng trong hình 5.9a, ta thấy rằng việc tăng luồng làm tăng giá trị luồng trên các cung (1,2), (2,4), 3,5), (5,6) và giảm luồng trên cung (3,4).

Cuối cùng, giả sử rằng ta đã nhận được các giá trị luồng thặng dư. Làm thế nào để các định x_{ij} ? Vì $r_{ij} = u_{ij} - x_{ij} + x_{ji}$, có rất nhiều tổ hợp của x_{ij} và x_{ji} tương ứng với cùng một giá trị r_{ij}. Ta có thể chọn một trong những khả năng này. Nhận xét rằng, ta có thể viết lại biểu thức $r_{ij} = u_{ij} - x_{ij} + x_{ji}$ thành $x_{ij} - x_{ji} = u_{ij} - r_{ij}$. Như vậy, ta có thể chọn $x_{ij} = u_{ij} - r_{ij}$ và $x_{ji} = 0$ nếu $u_{ij} - r_{ij} > 0$; ngược lại ta chọn $x_{ij} = 0$ và $x_{ji} = r_{ij} - u_{ij}$.

5.3.2. Ảnh hưởng của phép tăng trưởng lên sự phân rã luồng

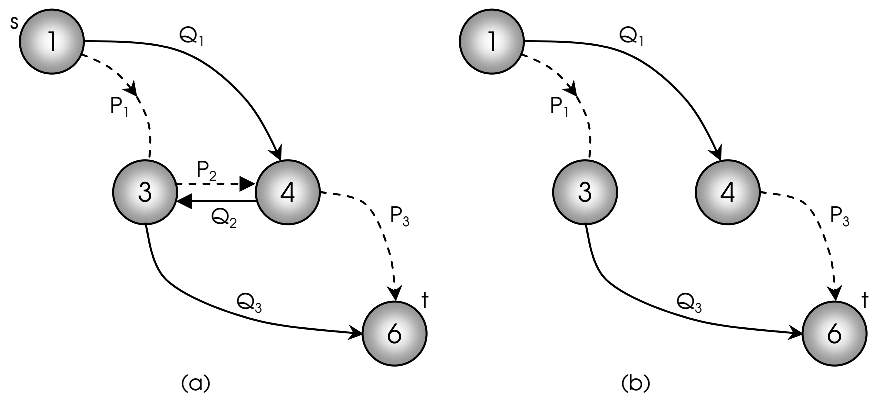
Để hiểu rõ hơn về thuật toán đường tăng trưởng, chúng ta hãy xem minh họa về hiệu ứng của sự tăng trưởng tác động lên sự phân rã luồng trong ví dụ ở hình 5.9. Hình 5.10a cho chúng ta thấy sự phân rã của luồng ban đầu và hình 5.10b cho chúng ta thấy sự phân rã của luồng sau khi ta đã tăng 1 đơn vị luồng dọc theo đường đi 1-2-4-3-5-6. Mặc dù chúng ta tăng 1 đơn vị luồng dọc theo đường đi 1-2-4-3-5-6 nhưng trong phân rã luồng lại không chứa con đường này. Tại sao ?



(a) (b)

Hình 5.10 Kết quả của phân rã luồng

Con đường 1-3-4-6 xác định luồng trong hình 5.10 chứa 3 đoạn: đoạn đường đến nút 3, cung (3,4) là 1 cung tới, và đoạn đường tới nút 6. Ta có thể xem con đường này như một sự tăng trưởng 0 đơn vị luồng. Tương tự như vậy, đường 1-2-4-3-5-6 chứa 3 đoạn: đoạn đường đến nút 4, cung (3,4) là 1 cung lui, và đoạn đường tới nút 6. Ta có thể xem sự tăng trưởng trên đường 1-2-4-3-5-6 như là sự liên kết đoạn đầu của con đường 1-3-4-6 với đoạn cuối của đường tăng trưởng và nối đoạn cuối của đường 1-3-4-6 với đoạn đầu của đường tăng trưởng (1-2-4-3-5-6) đồng thời triệt tiêu giá trị luồng trên cung



Hình 5.11 Ảnh hưởng của phép tăng trưởng lên sự phân rã luồng: (a) hai đường tăng trưởng P_1-P_2-P_3 va Q_1-Q_2-Q_3 ; (b) Ảnh hưởng thật sự của chúng

(3, 4) (xem hình 5.11).

5.4. Thuật toán gán nhãn và định lý lát cắt nhỏ nhất

Trong mục này, chúng ta sẽ khảo sát thuật toán đường tăng trưởng một cách chi tiết hơn. Trong mục trước, khi khảo sát thuật toán này, chúng ta đã bỏ qua một số chi tiết quan trọng, chẳng hạn như (1) làm thế nào để tìm ra một đường tăng trưởng hoặc khẳng định rằng mạng không chứa đường tăng trưởng nào, và (2) liệu thuật toán có dừng sau một số hữu hạn lần lặp không và dừng ở đâu, liệu thuật toán có thật sự tìm ra luồng cực đại như mong muốn hay không ? Trong mục này, chúng ta sẽ xem xét các vấn đề trên với một phiên bản cài đặt của thuật toán đường tăng trưởng được biết với tên gọi thuật toán gán nhãn (*labeling algorithm*).

Thuật toán gán nhãn sử dụng một kỹ thuật tìm kiếm (theo chiều rộng hoặc theo chiều sâu) để tìm một đường đi từ nút nguồn s đến nút đích t trong mạng G(x). Thuật toán sẽ lần đến tất cả các nút có thể đến từ nút nguồn theo một đường đi có hướng. Tại mỗi bước, thuật toán phân hoạch tập các nút của mạng thành hai nhóm: nhóm các nút được gán nhãn và không được gán nhãn. Các nút được gán nhãn là những nút mà tại thời điểm đang xét, thuật toán đã phát hiện ít nhất một đường đi nối từ nút nguồn đến nút này. Những nút còn lại là những nút không được gán nhãn. Thuật toán lập đi lập lại quá trình chọn một nút đã có nhãn và quét theo danh sách các nút kề với nó để gán nhãn cho các nút chưa có nhãn. Quá trình này được tiến hành cho đến khi nút đích t được gán nhãn. Khi đó, ta tìm được một đường đi từ s đến t. Đây chính là một đường tăng trưởng. Gửi một lượng luồng tối đa có thể dọc theo con đường này ta sẽ tăng được giá trị luồng x trong mạng ban đầu. Với mạng G(x) mới, ta tiếp tục quá trình gán nhãn và tăng luồng cho đến khi không thể gán nhãn được cho nút đích t. Thuật toán dừng tại đây. Dưới đây là thuật toán gán nhãn:

```
Thuật toán Labeling
Begin
   repeat
       Đặt mọi nút ở trạng thái không gán nhãn;
       Với mọi nút j ∈ N, đặt prev(j) = 0;
       Gán nhãn cho nút s và đặt LIST = {s};
       while (LIST ≠ {} và t chưa có nhãn)
       begin
           Lấy một nút I từ LIST;
           for (mỗi cung (i, j) trong mạng thặng dư) do
               if (r_ij > 0 và nút j chưa có nhãn) then
               begin
                   Đặt prev(j) = i; Gán nhãn cho j;
                   Thêm j vào LIST;
               end;
       end;
```

if (t được gán nhãn) then Tăng luồng;
until (t không có nhãn);
End;

Procedure Tăng luồng
Begin
Dùng giá trị nhãn prev() để tìm đường tăng trưởng P từ s đến t;
$\delta = \min\{r_{ij}: (i,j) \in P\}$;
Tăng δ đơn vị luồng dọc theo P và cập nhật khả năng thông qua thặng dư trong mạng thặng dư;
End;

5.4.1. Tính đúng đắn của thuật toán gán nhãn

Để khảo sát tính đúng đắn của thuật toán gán nhãn, ta cần chú ý rằng trong mỗi lần lặp của vòng lặp repeat, thuật toán hoặc tìm ra một đường tăng trưởng hoặc không gán nhãn được cho nút đích t. Trong trường hợp sau, ta phải chứng minh giá trị của luồng x là giá trị luồng cực đại. Giả sử tại giai đoạn này, S là tập các nút đã được gán nhãn và $\bar{S} = N - S$ là tập các nút không được gán nhãn. Rõ ràng, $s \in S$ và $t \in \bar{S}$. Do thuật toán không thể gán nhãn cho bất kỳ nút nào trong $\bar{S}$ nên $r_{ij} = 0$ với mọi $(i,j) \in (S, \bar{S})$. Hơn nữa, do $r_{ij} = (u_{ij} - x_{ij}) + x_{ji}$, $x_{ij} \le u_{ij}$ và $x_{ji} \ge 0$, nên $x_{ij} = u_{ij}$ với mọi $(i,j) \in (S, \bar{S})$ và $x_{ij} = 0$ với mọi $(i,j) \in (\bar{S}, S)$. Thay các giá trị này vào biểu thức (5.5) ta có:

$$v = \sum_{(i,j)\in(S,\bar{S})} x_{ij} - \sum_{(i,j)\in(\bar{S},S)} x_{ij} = \sum_{(i,j)\in(S,\bar{S})} u_{ij} = u[S,\bar{S}].$$

Như vậy, giá trị của luồng x hiện tại bằng khả năng thông qua của lát cắt $[S, \bar{S}]$. Kết hợp với mệnh đề 5.1, ta có thể khẳng định x chính là một luồng cực đại và $[S, \bar{S}]$ là lát cắt nhỏ nhất. Kết luận này chứng minh tính đúng đắn của thuật toán và nó cũng chứng minh luôn định lý về lát cắt nhỏ nhất.

Định lý 5.2: (Định lý về lát cắt nhỏ nhất – Ford-Fullkerson)

Giá trị cực đại của luồng gửi từ nút nguồn s đến nút đích t trên mạng bằng khả năng thông qua của lát cắt s-t nhỏ nhất.

Phần chứng minh của định lý về lát căt nhỏ nhất chỉ ra rằng khi thuật toán gán nhãn dừng, ta xác định được cả lát cắt nhỏ nhất. Thuật toán này cũng chứng minh định lý về đường tăng trưởng sau:

Định lý 5.3: (Định lý về đường tăng trưởng)

Một luồng x là luồng cực đại khi và chỉ khi mạng thặng dư G(x*) không chứa đường tăng trưởng.*

Chứng minh:

Nếu mạng thặng dư G(x*) chứa một đường tăng trưởng, rõ ràng luồng x* không thể là luồng cực đại. Ngược lại, nếu mạng thặng dư G(x*) không chứa đường tăng trưởng, tập nút S được gán nhãn bởi thuận toán gán nhãn xác định một lát cắt $[S, \overline{S}]$ có khả năng thông qua bằng với độ lớn của luồng x*. Từ đây suy ra x* là một luồng cực đại. Định lý được chứng minh.

Thuật toán gán nhãn còn cho ta một kết quả quan trọng hơn.

Định lý 5.4: (Định lý về tính nguyên)

Nếu khả năng thông qua của tất cả các cung trên mạng đều là số nguyên, thì luồng cực đại sẽ có giá trị nguyên.

Chứng minh:

Ta chứng minh bắng qui nạp theo số lần tăng luồng. Do thuật toán gán nhãn bắt đầu từ luồng có giá trị 0 và khả năng thông qua của tất cả các cung trên mạng đều là số nguyên nên trên mạng thặng dư ban đầu sẽ chứa toàn các cung với độ thông qua nguyên. Trong mỗi lần lặp, luồng sẽ được tăng một lượng bằng độ thông qua thặng dư nhỏ nhất ứng với một cung nào đó trên đường tăng trưởng. Theo giả thiết qui nạp, giá trị này nguyên. Như vậy, trong lần lặp tiếp theo, độ thông qua của các cung trên mạng thặng dư sẽ lại nguyên. Do u_{ij} và r_{ij} đều nguyên nên x_{ij} cũng sẽ nguyên. Sau mỗi lần lặp, giá trị của luồng x tăng lên ít nhất một đơn vị, và giá trị của x không thể vượt quá khả năng thông qua của một lát cắt bất kỳ nên thuật toán sẽ dừng sau hữu hạn bước. Định lý được chứng minh.

Kết quả của định lý 5.4 không có nghĩa là luồng cực đại trên mạng bất kỳ là số nguyên. Bài toán luồng cực đại có thể có đáp số không nguyên.

5.4.2. Độ phức tạp của thuật toán gán nhãn

Ta sẽ khảo sát độ phức tạp của thuật toán trong trường hợp xấu nhất. Nhắc lại, sau mỗi lần lặp, ngoại trừ lấn cuối cùng, thuật toán thực hiện một lần tăng luồng. Dễ dàng thấy, mỗi lần tăng luồng ta tốn chi phí O(m). Vấn đề còn lại là xác định tối đa có bao nhiêu lần tăng luồng. Nếu tất cả các cung đều có độ thông qua nguyên ≤ U, độ thông qua của lát cắt (s, N-{s}) không vượt quá nU. Như vậy, giá trị luồng cực đại phải không vượt quá nU. Thuật toán tăng luồng ít nhất một đơn vị sau mỗi lần lặp. Vì vậy, số lần tăng luồng không vượt quá nU. Từ đây suy ra, độ phức tạp của thuật toán là O(nmU).

Định lý 5.5:

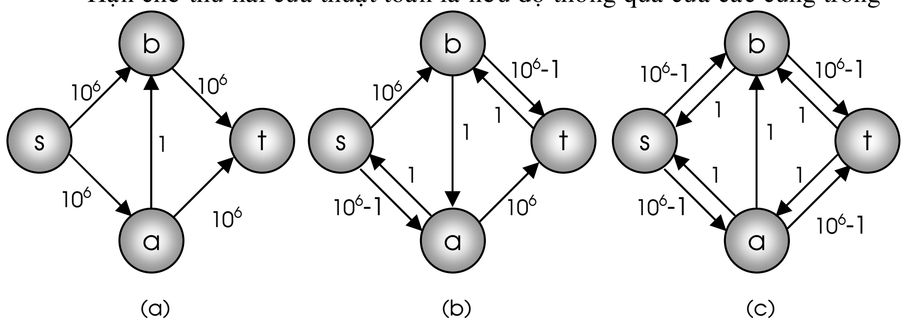
Thuật toán gán nhãn giải bài toán luồng cực đại với độ phức tạp O(nmU).

Mặc dù trong phần này, chúng ta giả thiết rằng độ thông qua của các cung đều hữu hạn. Trong một số ứng dụng, để thuận tiện cho việc mô hình hóa bài toán, ta cần cho phép các cung có độn thông qua vô hạn. Nếu ta giả sử rằng có một lát cắt s-t nào đó có độ thông qua hữu hạn và gọi U là khả năng thông qua lớn nhất của các cung trong lát cắt. Lúc đó, kết quả của định lý 5.5 cũng như các kết quả khác trong phần này vẫn bảo toàn tính đúng đắn. Một hướng tiếp cận khác là dùng một giá trị đủ lớn để biểu diễn các giá trị vô hạn sao cho không ảnh hưởng đến giá trị của luồng cực đại.

5.4.3. Hạn chế của thuật toán gán nhãn

Thuật toán gán nhãn có thể là thuật toán đơn giản nhất dùng để giải bài toán luồng cực đại trên mạng. Trong thực nghiệm, thuật toán hoạt động đủ tốt. Tuuy nhiên, trường hợp xấu nhất của thuật toán sẽ không thỏa mãn yêu cầu về tốc độ khi U quá lớn. Ví dụ, nếu $U = 2^n$, chi phí của thuật toán sẽ là hàm mũ của số nút trong mạng. Hơn nữa, thuật toán sẽ thực hiện rất nhiều lần lặp như ví dụ trong hình 5.12.

Hạn chế thứ hai của thuật toán là nếu độ thông qua của các cung trong



Hình 5.12 Ví dụ khi thuật toán gán nhãn làm việc không tốt: (a) mạng thặng dư ứng với luồng zero; (b) mạng sau khi tăng 1 đơn vị luồng dọc theo đường s-a-b-t; (c) mạng sau khi tăng 1 đơn vị luồng dọc theo đường s-b-a-t

mạng là số vô tỉ, thuật toán có thể không dừng (xem trong phần bài tập). Vì vậy, để bảo đảm tính hiệu quả của thuật toán, ta cần chọn đường tăng trưởng một cách cẩn thận.

Hạn chế thứ ba của thuật toán là *sự lãng phí*. Trong mỗi bước lặp, ta thực hiện qua trình gán nhãn các nút từ đầu, trong khi trong nhiều trường hợp các thông tin về nhãn của bước trước vẫn còn giá trị ở lần lặp sau. Việc giữ lại các thông tin này sẽ làm giảm đáng kể chi phí thực hiện thuật toán. Các bạn hãy thử cải tiến thuật toán theo hướng này như một bài tập.

5.4.4. Thuật toán Ford-Fullkerson

Trong phần trước, chúng ta khảo sát thuật toán tìm luồng cực đại theo hướng tiếp cận dùng mạng thăng dư. Ta có thể giải quyết bài toán này theo hướng tiếp cận khác. Ta có một biến thể của thuật toán gán nhãn. Thuật toán này sẽ làm việc trực tiếp trên mạng ban đầu.

Gọi cung (i, j) là cung bão hòa nếu giá trị luồng trên cung này $x_{ij} = u_{ij}$. Một đường đi từ s đến t là một dãy các cung không quan tâm chiều nối s với t (dây chuyền). Các cung hướng theo chiều từ s đến t gọi là cung thuận và ngược chiều ta gọi là cung nghịch.

Ta cũng định nghĩa khả năng thông qua thặng dư của một cung (i, j) trong đường đi P từ s đến t như sau:

$$r_{ij} = \begin{cases} u_{ij} - x_{ij} & \text{neáulaø(i, j) cung thuaäntreânP} \\ x_{ij} & \text{neáulaø(i, j) cung nghòch treânP} \end{cases}$$

Ta cũng định nghĩa khả năng thông qua thặng dư δ của một đường đi P như sau:

$\delta(P) = \min\{r_{ij}: (i,j) \in P\}$

Thuật toán Ford-Fullkerson

```
Begin
    Khởi động luồng x = 0;
    repeat
        /*Tìm một đường đi P từ s đến t trong đồ thị bộ phận gồm
        các cung không bão hòa của G*/
        Đặt mọi nút ở trạng thái không gán nhãn;
        Gán nhãn 0 cho nút s và đặt LIST = {s};
        while (LIST ≠ {} và t chưa có nhãn)
        begin
            Lấy một nút i từ LIST;
            for (mỗi cung (i, j) ∈ A) do
                if (u_ij - x_ij > 0 và nút j chưa có nhãn) then
                begin
                    Gán nhãn +i cho j;
                    Thêm j vào LIST;
                end;
            for (mỗi cung (j, i) ∈ A) do
                if (x_ji > 0 và nút j chưa có nhãn) then
```

```
                begin
                      Gán nhãn -i cho j;
                      Thêm j vào LIST;
                end;
        end;
        if (t được đánh dấu) then /*tăng luồng*/
        begin
              Tìm đường đi P từ s đến t dựa theo nhãn của các đỉnh (đỉnh
              có nhãn <0 sẽ cho ra cung ngược trên P).
              δ = khả năng thông qua thặng dư của P;
              tăng luồng x_ij += δ với mọi (i,j) ∈ P định hướng thuận;
              giảm luồng x_ij -= δ với mọi (i,j) ∈ P định hướng nghịch;
        endl
    until (t không được đánh dấu);
End;
```

5.5. Bài toán luồng tương thích (feasible flows problem)

Bài toán luồng tương thích là bài toán xác định một luồng trên mạng G=(N,A) thỏa mãn các ràng buộc sau:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b(i) \qquad \forall i \in N \qquad (5.10a)$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (i, j) \in A \qquad (5.10b)$$

trong đó $\sum_{i\in N} b(i) = 0$.

Bài toán luồng tương thích có một số ứng dụng thực tế. Chẳng hạn ứng dụng về phân phối hàng hóa. Trên một mạng lưới các hải cảng, một số cảng có hàng hóa mà những cảng khác cần. Ta biết lượng hàng có tại các cảng, biết nhu cầu về các hàng hóa này cũng như biết khả năng vận chuyển tối đa trên mỗi tuyến đường. Ta cần biết liệu có thể đáp ứng mọi nhu cầu dựa trên các nguồn cung cấp trong mạng hay không.

Ta có thể giải bài toán luồng tương thích bằng cách tìm luồng cực đại trên một mạng G' định nghĩa từ G như sau. Ta thêm vào hai nút mới, một nút nguồn s và một nút đích t. Với mỗi nút i có b(i)>0, ta thêm vào cung (s,

i) với độ thông qua là b(i), và với mỗi nút i có b(i)<0, ta thêm vào cung (i, t) với độ thông qua là -b(i). Sau khi thiết lập xong mạng G', ta giải bài toán luồng cực đại truyền từ s đến t trên mạng này. Nếu tất cả các cung (s, i) và (j, t) đều bão hòa (nghĩa là $x_{si} = u_{si}$ và $x_{jt} = u_{jt}$) thì bài toán (5.10) có luồng tương thích; trong trường hợp ngược lại, không có luồng tương thích.

Dễ dàng kiểm chứng tính đúng đắn của thuật toán. Nếu x là một luồng thỏa (5.10a) và (5.10b) thì chính luồng này với $x_{si} = b(i)$ trên mỗi cung (s, i) và với $x_{jt} = -b(i)$ trên mỗi cung (j, t) là luồng cực đại trên mạng G'. Tương tự như vậy, nếu x là luồng cực đại trên mạng G' và làm đầy tất cả các cung (s, i), (j, t) thì chính luồng này trong mạng G ban đầu sẽ thỏa (5.10).

5.6. Luồng có chặn dưới (Flows with lower bounds)

Trong phần này, chúng ta sẽ khảo sát bài toán luồng cực đại trên mạng với các cung có khả năng thông qua tối thiểu lớn hơn không (mạng có chặn dưới). Bài toán có thể phát biểu một cách hình thức như sau:

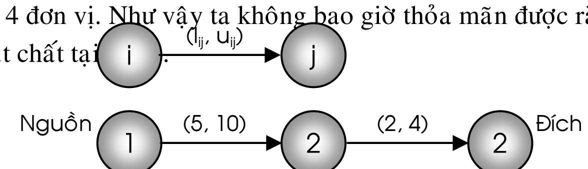
Cực đại hóa v

thỏa mãn ràng buộc:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} v & i = s, \\ 0 & \forall i \in N - \{s,t\} \\ -v & i = t \end{cases}$$

$$0 \le l_{ij} \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in A.$$

Trong mục trước, chúng ta đã khảo sát trường hợp đặc biệt của bài toán này khi tất cả $l_{ij} = 0$. Trong khi bài toán luồng trên mạng có chặn dưới bằng 0 luôn luôn có lời giải thích hợp, bài toán luồng trên mạng có chặn dưới lớn hơn 0 có thể không có lời giải thích hợp. Ví dụ, xem bài toán luồng cực đại trên mạng trong hình 5.13. Bài toán này không có lời giải vì cung (1,2) phải mang tối thiểu 5 đơn vị luồng trong khi cung (2,3) lại chỉ có thể mang tối đa 4 đơn vị. Như vậy ta không bao giờ thỏa mãn được ràng buộc về cân bằng vật chất tại [illegible].



Hình 5.13 Bài toán luồng cực đại không có luồng tương thích

Như đã minh họa trong ví dụ trên, mọi thuật toán tìm luồng cực đại trên mạng có chặn dưới phải giải quyết hai vấn đề: (1) xác định xem bài toán có tồn tại luồng tương thích hay không, và (2) nếu có, thì thiết lập luồng cực đại. Chính vì vây, không có gì ngạc nhiên khi hầu hết các thuật toán bao gồm hai giai đoạn. Giai đoạn thứ nhất kiểm tra sự tồn tại của một luồng tương thích và giai đoạn hai, nếu tồn tại sẽ biến đổi luồng tương thích thành luồng cực đại. Phần tiếp theo sẽ cho chúng ta thấy rằng mỗi giai đoạn của thuật toán sẽ tương đương với một lần giải bài toán luồng cực đại không có chặn dưới. Nghĩa là để giải bài toán luồng trên mạng có chặn dưới ta cần giải hai bài toán bài toán luồng trên mạng không có chặn dưới. Để thuận tiện, chúng ta sẽ khảo sát giai đoạn hai trước.

5.6.1. Xác định luồng cực đại

Giả sử ta có một luồng tương thích x trong mạng. Ta có thể hiệu chỉnh một thuật toán tìm luồng cực đại trên mạng không có chặn dưới bất kỳ để tìm luồng cực đại trên mạng của bài toán ta đang xét. Ta chỉ cần thực hiện hiệu chỉnh sau: ta định nghĩa độ thông qua thặng dư của cung (i,j) là $r_{ij} = (u_{ij} - x_{ij}) + (x_{ji} - l_{ji})$; hạng thức đầu tiên trong biểu thức thể hiện khả năng tăng tối đa của luồng chạy từ i đến j thông qua cung (i,j), và hạng thức thứ hai thể hiện khả năng tăng tối đa của luồng chạy từ i đến j thông qua việc giảm luồng trên cung (j,i). Lưu ý rằng x là luồng tương thích nên $r_{ij} \geq 0$. Do thuật toán tìm luồng cực đại trình bày trong phần trước chỉ làm việc trên mạng thặng dư nên ta có thể dung nó để thiết lập luồng cực đại trên mạng. Thuật toán này dừng khi ta nhận được giá trị luồng thặng dư tối ưu. Từ các giá trị luồng thặng dư này, ta có thể tính ra luồng cực đại bằng nhiều cách khác nhau. Ví dụ, ta có thể đổi biến số để giảm chi phí tính toán. Đặt $u'_{ij} = u_{ij} - l_{ij}$, $r'_{ij} = r_{ij}$, và $x'_{ij} = x_{ij} - l_{ij}$. Do $r_{ij} = (u_{ij} - x_{ij}) + (x_{ji} - l_{ji})$ nên $r'_{ij} = u'_{ij} - x'_{ij} + x'_{ji}$. Tương tự, $r'_{ji} = u'_{ji} - x'_{ji} + x'_{ij}$. Tính x' theo u' và r' ta được:

$x'_{ij} = \max(u'_{ij} - r'_{ij}, 0)$ và $x'_{ji} = \max(u'_{ji} - r'_{ji}, 0)$

Từ đây, ta có thể tính x theo công thức

$$x_{ij} = l_{ij} + \max(u_{ij} - r_{ij} - l_{ij}, 0)$$

$$x_{ji} = l_{ji} + \max(u_{ji} - r_{ji} - l_{ji}, 0)$$

Bây giờ chúng ta sẽ chứng minh rằng luồng cho bởi thuật toán đã hiệu chỉnh trên thật sự cho ta luồng cực đại trên mạng có chặn dưới. Gọi x là một luồng tương thích trên mạng G với giá trị luồng là v, và $[S, \overline{S}]$ là một lát cắt s-t. Ta định nghĩa độ thông qua của một lát cắt s-t như sau:

$$u[S,\overline{S}] = \sum_{(i,j)\in(S,\overline{S})} u_{ij} - \sum_{(i,j)\in(\overline{S},S)} l_{ij} \tag{5.11}$$

Độ thông qua của lát cắt cho biết lượng luồng thật sự có thể gửi từ các nút trong tập S đến các nút trong tập $\overline{S}$. Ta viết lại biểu thức (5.5) để tiện sử dụng:

$$v = \sum_{(i,j)\in(S,\overline{S})} x_{ij} - \sum_{(i,j)\in(\overline{S},S)} x_{ij} \tag{5.12}$$

Do $l_{ij} \le x_{ij} \le u_{ij}$ nên thay x_{ij} bằng u_{ij} trong tổng thứ nhất và thay xij bằng l_{ij} trong tổng thứ hai trong vế phải của (5.12) ta được bất đẳng thức:

$$v \le \sum_{(i,j)\in(S,\overline{S})} u_{ij} - \sum_{(i,j)\in(\overline{S},S)} l_{ij} = u[S,\overline{S}] \tag{5.13}$$

Bất đẳng thức (5.13) chứng tỏ rằng giá trị luồng cực đại nhỏ hơn hay bằng khả năng thông qua của một lát cắt s-t bất kỳ. Khi kết thúc, thuật toán cho ta một lát cắt $[S, \overline{S}]$ với $r_{ij} = 0$ trên mọi cung $(i,j) \in (S, \overline{S})$. Gọi x là luồng tương ứng, có giá trị v. Vì $r_{ij} = (u_{ij} - x_{ij}) + (x_{ji} - l_{ji})$, điều kiện $x_{ij} \le u_{ij}$ và $l_{ji} \le x_{ji}$ cho phép ta khẳng định $x_{ij} = u_{ij}$ và $l_{ji} = x_{ji}$. Như vậy, $x_{ij} = u_{ij}$ với mọi cung $(i,j) \in (S, \overline{S})$ và $l_{ij} = x_{ij}$ với mọi cung $(i,j) \in (\overline{S}, S)$. Thay các giá trị này vào (5.12) ta được

$$v = u[S,\overline{S}] = \sum_{(i,j)\in(S,\overline{S})} u_{ij} - \sum_{(i,j)\in(\overline{S},S)} l_{ij} \tag{5.14}$$

Từ (5.13) và (5.14) ta có thể kết luận, $[S, \overline{S}]$ là lát cắt s-t nhỏ nhất và x là luồng cực đại. Như là hệ quả của kết quả này, ta có định lý mở rộng của định lý lát cắt nhỏ nhất:

Định lý 5.5: (Định lý lát cắt nhỏ nhất mở rộng)

Nếu khả năng thông qua của một lát cắt s-t [S, $\overline{S}$] trên mạng có cả chặn trên lẫn chặn dưới định nghĩa như trong 5.12, thì luồng cực đại gửi từ s đến t sẽ có giá trị bằng khả năng thông qua của lát cắt nhỏ nhất.

5.6.2. Xây dựng luồng tương thích

Bây giờ, công việc còn lại là xá định một luồng tương thích trên mạng. Trước tiên, chúng ta sẽ chuyển đổi bài toán luồng cực đại sang bài toán lưu thông bằng cách thêm vào mạng cung (t, s) với khả năng thông qua vô hạn ∞. Cung này sẽ chuyển tải toàn bộ luồng gửi từ s đến t quay ngược trở lại s. Như vậy, trong bài toán lưu thông, luồng đi ra từ một nút bất kỳ (kể cả các nút s và t) bằng đúng luồng vào nó. Dễ dàng thấy rằng, bài toán luồng cực đại có luồng tương thích khi và chỉ khi bài toán lưu thông thương ứng có luồng tương thích. Bây giờ, ta sẽ tìm điều kiện cần và đủ để bài toán lưu thông trên mạng có chặn trên và chặn dưới có luồng tương thích.

Bài toán xác định luồng tương thích của bài toán lưu thông là bài toán tìm một luồng x thỏa các ràng buộc sau:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = 0 \qquad \forall i \in N \qquad (5.15a)$$

$$l_{ij} \le x_{ij} \le u_{ij} \qquad \forall (i, j) \in A \qquad (5.15b)$$

Bằng cách thay $x_{ij} = x'_{ij} + l_{ij}$ vào ràng buộc (5.15a) và (5.15b), ta nhận được bài toán mới tương đương:

$$\sum_{\{j:(i,j)\in A\}} x'_{ij} - \sum_{\{j:(j,i)\in A\}} x'_{ji} = b(i) \qquad \forall i \in N \qquad (5.16a)$$

$$0 \le x'_{ij} \le u_{ij} - l_{ij} \qquad \forall (i, j) \in A \qquad (5.16b)$$

với giá trị cung/cầu b(.) tại mỗi nút xác định bởi công thức:

$$b(i) = \sum_{\{j:(j,i)\in A\}} l_{ji} - \sum_{\{j:(i,j)\in A\}} l_{ij} = 0 \qquad \forall i \in N \qquad (5.17)$$

Nhận xét rằng $\sum_{i \in N} b(i) = 0$ do mỗi l_{ij} xuất hiện 2 lần trong biểu thức này, một lần với dấu cộng và một lần với dấu trừ. Như vậy, bài toán lưu thông tương đương với việc tìm luồng x' thỏa mãn ràng buộc (5.16).

Lưu ý rằng, bài toán này chính là bài toán luồng tương thích chúng ta đã khảo sát ở phần trước. Như đã biết, để giải bài toán luồng tương thích ta cần giải một bài toán luồng cực đại trên mạng không có chặn dưới. Qua đó, hoặc ta xác định được một lời giải thỏa (5.16) hoặc khẳng định được rằng, bài toán không tồn tại lời giải phù hợp. Nếu x'_{ij} là luồng tương thích của (5.16), thì $x_{ij} = x'_{ij} + l_{ij}$ là luồng tương thích của (5.15).

5.6.3. Mô tả đặc điểm của luồng tương thích trên mạng lưu thông

Ta sẽ khảo sát sự tồn tại luồng tương thích của bài toán lưu thông. Gọi S là một tập hợp nút bất kỳ trên mạng. Ta có:

$$\sum_{(i,j)\in(S,\bar{S})} x_{ij} - \sum_{(i,j)\in(\bar{S},S)} x_{ij} = 0 \qquad (5.18)$$

Do $l_{ij} \le x_{ij} \le u_{ij}$, từ (5.18) suy ra:

$$\sum_{(i,j)\in(\bar{S},S)} l_{ij} \le \sum_{(i,j)\in(S,\bar{S})} u_{ij} \qquad (5.19)$$

Biểu thức (5.19) chính là điều kiện cần để tồn tại luồng tương thích. Nó có ý nghĩa là, lượng luồng tối đa có thể gửi ra từ tập S phải lớn hơn lượng luồng tối thiểu có thể nhận bởi tập S.

Ta sẽ đưa ra một chứng minh dạng thuật toán để chứng minh (5.19) là điều kiện đủ. Ta bắt đầu từ một luồng x thỏa các ràng buộc của bài toán ngoại trừ một số vi phạm về chặn dưới. Ta sẽ từng bước biến đổi x thành luồng tương thích hoặc phát hiện ta rằng một nút nào đó trong tập S vi phạm điều kiện (5.19).

Ứng với luồng x, ta gọi một cung (i, j) là không tương thích nếu $x_{ij} < l_{ij}$ và gọi là tương thích nếu $x_{ij} \ge l_{ij}$. Chọn một cung (p, q) không tương thích và sẽ biến nó thành tương thích bằng cách tăng luồng trên nó. Do điều kiện ràng buộc về cân bằng vật chất tại các nút, để tăng luông trên cung (p, q) ta phải tăng luồng dọc theo một hay nhiều chu trình chứa (p, q). Ta định nghĩa mạng thặng dư G(x) giống như trước đây, ngoại trừ việc khả năng thông qua thặng dư r_{ij} được đặt bằng u_{ij} - x_{ij}. Mọi chu trình tăng trưởng chứa cung (p, q) như một cung tới phải chứa một đường đi có hướng trong G(x) từ q đến p công với cung (p, q). Ta có thể dùng một thuật toán gán nhãn để xác định đường đi từ q đến p.

Ta áp dụng thủ tục này trên mỗi cung không tương thích một lần. Sau mỗi bước tính không tương thích của các cung sẽ giảm dần. Lặp lại qua trình này, hoặc ta sẽ nhận được một luồng tương thích hoặc thuật toán gán nhãn sẽ không thể tìm ra một đường đi từ q đến p đối với một cung không tương thích (p, q) nào đó. Trong trường hợp này, bài toán của chúng ta sẽ không có lời giải. Thật vậy, gọi S là tập nút đã được gán nhãn trong lần áp dụng cuối cùng thuật toán gán nhãn. Rõ ràng, $q \in S$ và $p \in \bar{S} = N - S$. Do thuật toán gán nhãn không thể gán nhãn cho các nút thuộc $\bar{S}$, nên mọi cung (i, j) hướng từ S đến $\bar{S}$ có khả năng thông qua thặng dư bằng không. Do đó, $x_{ij} = u_{ij}$ với mọi cung $(i, j) \in (S, \bar{S})$ và $x_{ij} \le u_{ij}$ với mọi cung $(i, j) \in (\bar{S}, S)$. Ta cũng có $(p, q) \in (\bar{S}, S)$ và $x_{pq} < u_{pq}$. Thay các giá trị này vào (5.18) ta có:

$$\sum_{(i,j)\in(\bar{S},S)} l_{ij} > \sum_{(i,j)\in(S,\bar{S})} u_{ij}$$

Kết quả này mâu thuẫn với điều kiện (5.19). Điều này có nghĩa là ta đã chứng minh (5.19) là điều kiện đủ cho sự tồn tại luồng tương thích. Từ đây, ta có định lý sau:

Định lý 5.6: (Điều kiện tồn tại luồng tương thích trên mạng lưu thông)

Trên mạng lưu thông có chặn dưới dương tồn tại luồng tương thích khi và chỉ khi mọi tập đỉnh S thỏa mãn điều kiện:

$$\sum_{(i,j)\in(\bar{S},S)} l_{ij} \le \sum_{(i,j)\in(S,\bar{S})} u_{ij}$$

Từ định lý 5.6, ta dễ dàng suy ra định lý về sự tồn tại lời giải của bài toán luồng tương thích đã trình bày trong mục 5.5.

Định lý 5.7: *Bài toán luồng tương thích (5.10) có lời giải khi và chỉ khi với mọi tập con các nút S,* $b(S) - u[S, \bar{S}] \le 0$, *trong đó* $b(S) = \sum_{i\in S} b(i)$.

6. BÀI TOÁN CẶP GHÉP

6.1. Bài toán cặp ghép trên đồ thị hai phía

6.2. Bài toán cặp ghép trên đồ thị hai phía với chi phí tối thiểu

6.3. Bài toán cặp ghép trên đồ thị tổng quát

TÓM TẮT

Trong chương này, chúng ta đã nghiên cứu một số khái niệm cơ bản về luồng trên mạng. Trong đó, đặc biệt khảo sát kỹ bài toán luồng cực đại và một số biến thể của nó. Sau khi minh họa một số ứng dụng, chúng ta đã chỉ ra rằng bài toán luồng cực đại và bài toán lát cắt nhỏ nhất quan hệ với nhau rất chặt chẽ và khi giải bài toán luồng cực đại ta cũng giải luôn được bài toán lát cắt nhỏ nhất. Kết quả này thể hiện qua *định lý lát cắt nhỏ nhất*. Từ định lý này, ta nhận được nhiều kết quả lý thú.

Tiếp theo, chúng ta khảo sát bài toán luồng cực đại với độ thông qua của các cung bị chặn dưới (> 0). Ta có thể giải quyết bài toán này theo hai giai đoạn. Giai đoạn đầu để xác định sự tồn tại của luồng tương thích và giai đoạn hai biến đổi luồng tương thích này thành luồng cực đại. Trong cả hai giai đoạn, ta đều phải giải một bài toán luồng cực đại trên mạng không có chặn dưới. Ta cũng khảo sát điều kiện cần và đủ để ttong tại luồng tương thích.

Cuối cùng, chúng ta khảo sát một bài toán có quan hệ gần gũi với bài toán luồng trên mạng là bài toán cặp ghép. Bài toán này có nhiều ứng dụng thực tế và một số dạng của nó có thể giải bằng cách giải bài toán luồng cực đại. Ngoài ra, chúng ta cũng khảo sát thuật toán Hungary dành riêng cho việc giải bài toán này. Do điều kiện hạn chế, chúng ta chỉ tập trung khảo sát bài toán cặp ghép trên đồ thị hai phía.

BÀI TẬP

1. **Bài toán ăn tối**: một số gia đình đi ăn tối chung với nhau. Để tăng mối quan hệ xã hội, họ muốn sắp đặt cách ngồi sao cho không có hai thành viên của cung một gia đình ngồi chung bàn. Hãy chỉ ra cách mô hình

hóa bài toán này như một bài toán luồng cực đại. Giả sử rằng, có tất cả p gia đình và gia đình thứ i có a(i) thành viên. Có tất cả q cái bàn và bàn thứ j có b(j) chỗ ngồi.

2. Hãy xây dựng một lớp các mạng mà trên đó số lượng lát cắt s-t tăng theo hàm mũ của n (e^n).

3. Hãy xây dựng một lớp các mạng mà trên đó số lượng lát cắt tối thiểu tăng theo hàm mũ của n (e^n).

4. Hãy xây dựng một thuật toán xác định lát cắt tối thiểu [S, $\overline{S}$] thỏa mãn điều kiện: nếu [R, $\overline{R}$] là lát cắt tối thiểu thì $R \subseteq S$.

5. Hãy xây dựng một thuật toán xác định lát cắt tối thiểu [S, $\overline{S}$] thỏa mãn điều kiện: nếu [R, $\overline{R}$] là lát cắt tối thiểu thì $S \subseteq R$.

6. **Bài toán luồng cực tiểu (minimum flow problem):** bài toán luồng cực tiểu có mối quan hệ rất gần với bài toán luồng cực đại có chặn dưới. Trong bài toán luồng cực tiểu, ta muốn gửi một lượng tối thiểu luồng từ nút nguồn s đến nút đích t thỏa mãn ràng buộc về chặn trên và chặn dưới của các cung.

a) Hãy chỉ ra cách giải bài toán trên bằng hai bài toán luồng cực đại không có chặn dưới. (*Gợi ý: trước tiên tìm luồng tương thích, sau đó chuyển luồng này thành luồng cực tiểu*)

b) Hãy chứng minh *định lý lát cắt cực đại* sau:

Nếu gọi l[S, $\overline{S}$] là chặn dưới của khả năng thông qua của lát cắt s-t, định nghĩa bởi công thức sau:

$$l\left[S,\overline{S}\right]= \sum_{(i,j)\in(S,\overline{S})} l_{ij} - \sum_{(i,j)\in(\overline{S},S)} u_{ij}$$

Giá trị luồng tối thiểu từ s đến t bằng giá trị chặn dưới l[S, $\overline{S}$] cực đại của các lát cắt.

7. **Mạng với các nút có độ thông qua:** trong một số mạng, ngoài khả năng thông qua của các cung, người ta còn gán cho mỗi nút i (khác nút nuồn s và nút đích t) độ thông qua w(i) cho biết lượng luồng tối đa có thể đi qua nút này. Ví dụ, trên mạng giao thông đường không, mỗi sân bay bị giới hạn bởi số đường băng để hạ và cất cánh; hoặc trên mạng máy tính, các switch hay HUB bị giới hạn bởi số lượng các cổng. Trên các mạng loại này, chúng ta cần tìm luồng cực đại thỏa mãn cả hai ràng buộc về khả năng thông qua của nút cũng như cung. Hãy xây dựng thuật toán giải bài toán trên.