# BÀI TOÁN LUỒNG VÀ ỨNG DỤNG TRONG THỰC TẾ

VŨ ĐÌNH HÒA và ĐOÀN PHỨC

## I – MỞ ĐẦU

Bài toán luồng là một bài toán rất thường gặp trong thực tiễn sản xuất và chiến đấu, chẳng hạn ta phải xem xét việc tổ chức một mạng lưới vận tải từ một địa điểm A đến một địa điểm B cho trước. Nhiều nhà toán học đã nghiên cứu và tìm ra nhiều thuật toán để xác định một luồng cực đại có thể có từ A tới B. Ngày nay bài toán luồng được xem xét dưới nhiều khía cạnh khác nhau, (chẳng hạn người ta khảo sát bài toán luồng với những hàm mục tiêu, hoặc hàm ràng buộc khác nhau). Ngoài ra người ta cũng tìm được rất nhiều bài toán xuất phát từ thực tiễn hoặc từ nghiên cứu lý thuyết có thể giải được nhờ thuật toán tìm luồng cực đại. Bài báo này nhằm giới thiệu bạn đọc một số những bài toán như vậy.

Bài toán luồng được phát biểu như sau:

Cho trước một đồ thị định hướng $G = [X, U]$ trên mỗi cạnh $u = [P, Q] \in U$ ta có một số thực $f(u)$. Như vậy một hàm số $f: U \to R$ được xác định, ở đây $R$ là tập các số thực.

Nếu tại một đỉnh P của G tổng số các giá trị của $f$ trên các cạnh đi vào P bằng tổng số các giá trị của $f$ trên các cạnh đi ra khỏi P (xem ví dụ ở hình 1) thì ta nói là $f$ thỏa mãn tại P định luật Kirchhoff. Một ví dụ trong kỹ thuật là dòng điện chạy qua một mạng điện nào đó thì tất cả các điểm của mạng đều thỏa mãn định luật Kirchhoff.

*Hình 1*

Trong bài toán luồng chúng ta không đòi hỏi tất cả các đỉnh của đồ thị đều thỏa mãn định luật Kirchhoff mà sẽ đánh dấu 2 đỉnh đặc biệt: một đỉnh Q được gọi là nguồn, còn đỉnh S được gọi là hạ lưu. Ngoài 2 đỉnh A và S nói trên, tất cả các đỉnh còn lại đều thỏa mãn định luật Kirchhoff. Ta có thể thấy ngoài ví dụ dòng điện, một mạng lưới vận chuyển từ nơi sản xuất Q tới nơi tiêu thụ S hoàn toàn thỏa mãn mô hình này. Một hàm số $f$ như nói trên thỏa mãn:

$$f(P,X) - f(X,P) = 0$$

cho mọi đỉnh $P \neq Q$ và $P \neq S$ của G. Trong đó ký hiệu $f(P,X)$ là tổng số các giá trị của $f$ trên các cạnh xuất phát từ P, còn $f(X,P)$ là tổng số các giá trị của $f$ trên các cạnh đi vào P. Khi đó ta gọi $f$ là *một dòng chảy* (hoặc là một luồng).

Một luồng $f$ được gọi là có cường độ v nếu

$$v = f(Q,X) - f(X\,Q).$$

Dễ thấy rằng ta cũng có $v = f(X,S) - f(S,X)$.

Một mạng vận tải là một bộ $N = [X; U; C; Q; S]$, ở đây Q là nguồn còn S là hạ lưu và C là một hàm số từ $U \to R$, với mỗi $u \in U$ thì $C(u)$ được gọi là khả năng thông qua của cạnh u. Một luồng $f$ của N được gọi là luồng chấp nhận được nếu có $0 \leqslant f(u) \leqslant C(u)$ cho mọi cạnh $u \in U$. Do ý nghĩa thực tế nên cho phép có thể có $C(u) = +\infty$.

*Bài toán luồng lớn nhất* là bài toán xác định trên mạng vận tải $N = \{X; U; C; Q; S\}$ một luồng chấp nhận được $f$ với một cường độ cực đại có thể.

## II – THUẬT TOÁN FORD-FULKERSON

Thuật toán Ford – Fulkerson là một thuật toán điều chỉnh, xuất phát từ một luồng $\varphi$ chấp nhận được trên N (ví dụ $\varphi(u) = 0$ cho mọi $u \in U$) với cường độ v, ta sẽ liên tục điều chỉnh sao cho ta thu được các luồng chấp nhận được mới, nhưng với cường độ lớn hơn.

Ta dán nhãn các đỉnh P của N bằng các vòng tròn nhỏ quanh các đỉnh và đặt các số $\varepsilon(P)$ vào trong các vòng tròn theo quy tắc sau:

(1) Dán nhãn Q và đặt $\varepsilon(Q) = +\infty$

(2) Nếu $u = [T,K] \in U$ là một cung: a) Nếu ở T đã có nhãn, còn K chưa có nhãn Khi đó ta dán nhãn cho K nếu

$$\varphi(u) < C(u),$$

và đặt $\varepsilon(K) = \min(\varepsilon(T), C(u) - \varphi(u))$.

b) T chưa có nhãn mà K đã có nhãn. Ta dán nhãn T nếu

$$\varphi(u) > 0$$

và đặt $\varepsilon(T) = \min(\varepsilon(K), \varphi(u))$.

(3) Thuật toán kết thúc nếu S được dán nhãn, hoặc là không một đỉnh nào còn có thể dán nhãn nữa.

Khi đó tồn tại một dãy cạnh đi từ Q tới S nếu như S được dán nhãn và ta có thể thay $\varphi$ bởi một dòng cho phép $\varphi^*$ khác theo công thức:

$\varphi^*(u) := \varphi(u) + \varepsilon(S)$ nếu u cùng hướng từ Q tới S

$\varphi^*(u) := \varphi(u) - \varepsilon(S)$ nếu u có hướng đi từ S tới Q,

cho các cạnh u ở trên đường đi từ Q tới S này, còn $\varphi^*(u) := \varphi(u)$ trên tất cả các cạnh khác. Dễ kiểm tra thấy là $\varphi^*$ là một luồng cho phép có cường độ lớn hơn cường độ của $\varphi$.

## III – MỘT SỐ BÀI TOÁN ĐƯỢC GIẢI BẰNG BÀI TOÁN LUỒNG

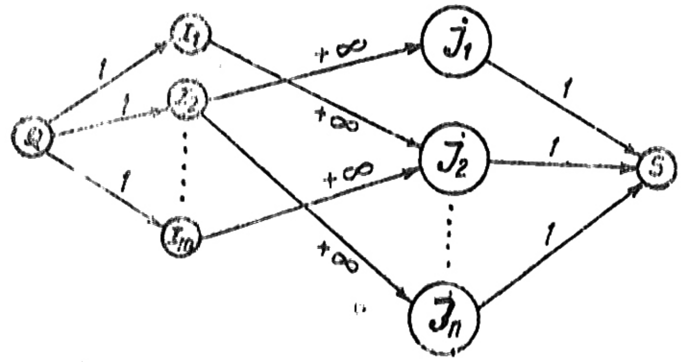
**1. Bài toán phân công việc (còn gọi là bài toán vợ chồng).**

Cho tập $I = \{I_1, I_2, \ldots, I_m\}$ gồm m người và tập $J = \{J_1, J_2, \ldots J_n\}$ gồm n công việc. Mỗi người đều biết làm một số việc nào đó.

*Bài toán 1*: Hãy sắp đặt m công việc của tập J cho m người sao cho mỗi người đều có một việc mà họ biết làm và mỗi việc được làm bởi nhiều nhất 1 người. Điều kiện cần để có thể có một cách xếp đặt thỏa mãn điều kiện này là nếu chọn ra một số k người tùy ý thì nhóm người này phải biết làm ít nhất k công việc. Và Konig và Hall đã chứng minh đây cũng là điều kiện đủ để cho ta có thể có một cách xếp đặt thỏa mãn yêu cầu bài toán.

Tuy vậy, định lý Konig – Hall không cho ta một cách sắp đặt cụ thể nào cả. Dựa vào thuật toán của Ford – Fulkerson ta có thể chỉ ra một cách phân công, nếu như bài toán có nghiệm.

Trước hết ta biểu diễn I và J thành các đỉnh của một đồ thị. Giữa $I_i$ và $J_j$ nào đó có một cung xuất phát từ $I_i$ tới $J_j$ chỉ khi người $I_i$ có thể làm công việc $J_j$ và đặt khả năng thông qua của mỗi cung này là $+\infty$.




Hình 2

Ngoài ra ta bổ sung thêm vào đồ thị này đỉnh nguồn Q và đỉnh hạ lưu S. Q được nối với tất cả các đỉnh của I, còn các đỉnh của J được nối với S. Khả năng thông qua của các cung mới này bằng 1. Như vậy dễ nhận thấy rằng cần và đủ để bài toán 1 giải được là luồng cực đại cho phép có cường độ m. Khi đó cùng với luồng cực đại này ta cũng có luôn một lời giải cho bài toán 1.

**2. Bài toán tìm số cực tiểu các dòng phủ của một ma trận.**

Cho trước một ma trận $A$ $(a_{ij})$, $i = 1, 2, \ldots, n$

$j = 1, 2, \ldots, m$.

Hai phần tử khác O của A được gọi là độc lập nếu như nó không cùng nằm trên một hàng hoặc một cột của ma trận.

Một hệ m phần tử khác O của A được gọi là một hệ độc lập nếu chúng đôi một độc lập với nhau.

Một tập hợp các dòng (tức là hàng hoặc cột) của ma trận được gọi là phủ A nếu như mỗi phần tử khác O của A nằm trên ít nhất 1 dòng của nó.

*Bài toán 2*: Hãy xác định số nhỏ nhất các dòng phủ một ma trận A cho trước.

Nhờ định lý Egervary khẳng định rằng số dòng nhỏ nhất phủ ma trận A đúng bằng số phần tử của một hệ độc lập có nhiều phần tử nhất, ta có thể giải bài toán 2 thông qua bài toán 3.

*Bài toán 3*: Hãy xác định số lớn nhất các phần tử $\neq$ O của A đôi một độc lập với nhau.

Bài toán này thực chất là bài toán 1 cho ứng với hai tập J và I là một ma trận A mà một phần tử $a_{ij} \neq 0$ khi và chỉ khi $I_i$ có thể làm việc $J_j$.

**3. Bài toán bố trí sản xuất trên một băng chuyền.**

Trên một khâu sản xuất song song (chẳng hạn lắp ráp một cái máy) có vị trí làm việc $J_1, J_2, \ldots J_n$ (ở mỗi vị trí làm một việc khác nhau chẳng hạn như phải lắp ráp một số chi tiết máy khác nhau vào máy) và có n người công nhân $I_1, I_2, \ldots, I_n$ và ta biết rằng thời gian hoàn thành một đơn vị công việc của người $I_i$ tại vị trí làm việc $J_j$ là $t_{ij}$ (do quá trình đào tạo tay nghề nên các đại lượng này rất phụ thuộc vào từng người công nhân). Tốc độ hoàn thành các công việc được tiến hành song song này phụ thuộc vào người làm chậm nhất (tuy nhiên anh ta có thể làm công việc khác nhanh hơn nhiều các đồng nghiệp của mình). Bài toán đặt ra:

*Bài toán 4*: Hãy bố trí công nhân vào vị trí làm việc để hoàn thành công việc một cách nhanh nhất.

Dạng toán học của bài toán 4 là như sau: Hãy xác định một hệ phủ R của ma trận $(t_{ij})$ $i = 1, 2, \ldots n$

$j = 1, 2, \ldots, n$

sao cho $\max\limits_{t_{ij} \in R} t_{ij}$ nhỏ nhất có thể được.

Thuật toán điều chỉnh như sau:

i) Ta chọn R là một hệ độc lập tùy ý. Đặt

$$\mu_0 = \max_{t_{ij} \in R} (t_{ij})$$

ii) Xác định ma trận $A = (a_{ij})$ theo quy tắc:

$$a_{ij} = \begin{cases} 1 \text{ nếu } t_{ij} < \mu_0 \\ 0 \text{ nếu } t_{ij} \geqslant \mu_0 \end{cases}$$

iii) Nếu bài toán (bài toán vợ chồng) tương ứng với ma trận A không giải được thì chính R là hệ cần tìm.

Còn nếu bài toán 1 giải được, thì tương ứng với lời giải của nó ta có hệ $R_1$ gồm n phần tử và $\mu_0 > \max\limits_{t_{ij} \in R} t_{ij} = \mu_1$

iv) Ta thay $R_1$ vào vị trí R và lại quay về i).

Có thể thấy rằng thuật toán bao giờ cũng kết thúc sau hữu hạn bước.

Với phương pháp mô hình hóa hoàn toàn tương tự, chúng ta có thể thấy rằng các bài toán sau cũng có thể giải bằng thuật toán Ford—Fulkerson.

**4. Bố trí lịch biểu trên sân bay.**

Giả sử ta có n sân bay, giả sử i và j là 2 sân bay khác nhau. Thời gian $t_{ij}$ là thời gian một chiếc máy bay từ sân bay i sang sân bay j. Trên mỗi sân bay i có một lịch biểu $F_i$ sau: Máy bay hạ cánh vào thời điểm $a_i$ và cất cánh vào thời điểm $b_i$ ($a_i \leqslant b_i$).

*Bài toán 5*: Tìm số máy bay tối thiểu để thỏa mãn các lịch biểu $F_i$.

**5. Lịch biểu trong máy tính có nhiều bộ xử lý.**

Cho tập $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ các mô đun và

tập $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ các bộ xử lý. Ta đặt

$f_{ik}$ = phí tổn thực hiện $M_i$ trên $P_k$

$$x_{ik} = \begin{cases} 1 \text{ nếu } M_i \text{ được thực hiện trên } P_k, \\ 0 \text{ trong trường hợp khác,} \end{cases}$$

$C_{ij} = C_{ji}$ = phí tổn liên lạc từ $M_i$ sang $M_j$.

*Bài toán 6*: Hãy tìm một phép phân phối

$X : \mathcal{M} \to \mathcal{P}$ sao cho:

$$\operatorname{cost}(X) := \sum_{k=1}^{m} \sum_{i=1}^{n} \left\{ f_{ik} x_{ik} + \sum_{l<k}^{n} \sum_{j<i}^{n} C_{ij} x_{ik} x_{jl} \right\}$$

nhỏ nhất có thể trong trường hợp cụ thể m = 2.

Nhận ngày 12-8-1987

### *TÀI LIỆU THAM KHẢO*


1. Sachs H., Einfühsung in die Theorie der endlichen Graphen. Leipzig 1970.

2. Berge C., Théorie des Graphes et ses applications. Paris 1967.

3. Wesley W. Chu, Leslie J. Holloway, Min-Tsung lan and Kemal Efe, « Task Allocation in distributed data Processing ». Computer, Now. 1980, p. 57–69, Vol. 13, Nr. 11–ISSN–0018–9162.

4. Harary. F., Proof techniques in Graph Theory. Newyork and London 1969.

5. Roberts F. Discrete Mathematical models. 1981. London – Sydney – Torontô – New Delhi – Tokyo – Singapore.



## ABSTRACT

### THE PROBLEM OF FLOW AND ITS APPLICATION

The algorithms for finding maximal netwook flows are used for finding solutions of some another problems.