

# THUẬT TOÁN SONG SONG TÌM LƯỜNG CỰC ĐẠI USING PARALLEL ALGORITHM TO FIND THE MAXIMAL FLOW

TRẦN QUỐC CHIẾN

*Trường Đại học Sư phạm, Đại học Đà Nẵng*

HỒ XUÂN BÌNH

*Trường Đại học Dân lập Duy Tân*

## TÓM TẮT

Kết quả chính của bài báo là tập trung xây dựng thuật toán song song dựa trên thuật toán truyền thống và thuật toán hoán chuyển nguồn đích [2], Các kết quả cơ bản được hệ thống và chứng minh. Ý tưởng thuật toán là sử dụng hai bộ vi xử lý thực hiện công việc song song tìm đường tăng luồng, vi xử lý 1 xuất phát từ đỉnh nguồn, vi xử lý 2 xuất phát từ đỉnh đích. Thuật toán song song làm giảm đáng kể thời gian tính toán so với các thuật toán truyền thống.

## ABSTRACT

The main result of this article is putting forward the parallel algorithm based on Ford-Fulkerson and source-sink alternative algorithm. The basic results are systematically presented and proved. The aim of the algorithm is using two processing to carry out parallelly working finding augmented paths, one comes from the source, and the other from the sink vertex. The proposed algorithm considerably decreases the computational time in comparison with non-parallel algorithms.

Key word: graph, network, flow, parallel

## 1. Đặt vấn đề

Bài toán luồng cực đại trên mạng là một trong số những bài toán tối ưu trên đồ thị tìm được những ứng dụng rộng rãi trong thực tế cũng như những ứng dụng thú vị trong lý thuyết tổ hợp. Bài toán được đề xuất và giải quyết bởi hai nhà toán học Mỹ Ford và Fulkerson vào đầu những năm 1950 và ngày càng được các nhà khoa học quan tâm nghiên cứu. Hiện nay, mô hình xử lý song song đã và đang phát triển mạnh mẽ giải quyết những vấn đề bế tắc mà mô hình xử lý tuần tự gặp phải như vấn đề thời gian thực hiện chương trình, tốc độ xử lý, khả năng lưu trữ của bộ nhớ, xử lý dữ liệu với quy mô lớn....

Trong bối cảnh đó, thuật toán tìm luồng cực đại cần được phát triển theo hướng song song nhằm phát huy sức mạnh của bài toán.

## 2. Bài toán tìm luồng cực đại trên mạng

Cho mạng  $G(V,E,C)$ , nguồn  $a$ , đích  $z$ . Trong số các luồng trên mạng  $G$ , hãy tìm luồng có giá trị lớn nhất.

## 3. Ý tưởng thuật toán

Dựa trên thuật toán truyền thống và thuật toán hoán chuyển nguồn đích, xây dựng thuật toán song song tìm luồng cực đại. Ý tưởng của phương pháp này là thay vì trong thuật toán truyền thống dùng một bộ vi xử lý thực hiện công việc tuần tự từ đỉnh nguồn đến đỉnh đích. Trong thuật toán song song sử dụng hai bộ vi xử lý thực hiện công việc song song, vi xử lý 1 xuất phát từ đỉnh nguồn, vi xử lý 2 xuất phát từ đỉnh đích. Hai vi xử lý trong quá trình tìm đường tăng luồng sẽ gặp nhau ở đỉnh trung gian  $t$  nào đó, công việc tiếp theo vi xử lý 1 xử lý công việc từ đỉnh  $t$  đến nút nguồn, vi xử lý 2 xử lý công việc từ đỉnh  $t$  đến nút đích.

## 4. Xây dựng thuật toán song song

► **Đầu vào:** Mạng  $G = (V, E)$  với nguồn  $a$ , đích  $z$ , khả năng thông qua:  $C = (c_{ij}), (i, j) \in G$ .

Các đỉnh trong G được sắp xếp theo thứ tự nào đó.

➤ **Đầu ra:** Luồng cực đại  $F = (f_{ij}), (i, j) \in G$ .

➤ **Các bước**

**Bước 1: Khởi tạo**

**P1:**

Luồng xuất phát: For  $i := 1$  to  $(n \text{ div } 2)$  do

For  $j := 1$  to  $n$  do if  $c_{ij} > 0$  then  $f_{ij} = 0$

Đặt nhãn tiến ( $\uparrow$ ) cho đỉnh nguồn:  $a(\uparrow, \phi, \infty)$

Tạo lập tập S gồm các đỉnh đã có nhãn tiến nhưng chưa được dùng để sinh nhãn tiến:  $S := \{a\}$ ; khởi gán điều kiện kết thúc  $\text{Stop} := \text{False}$ ;

**P2:**

Luồng xuất phát: For  $i := (n \text{ div } 2) + 1$  to  $n$  do

For  $j := 1$  to  $n$  do if  $c_{ij} > 0$  then  $f_{ij} = 0$

Đặt nhãn lùi ( $\downarrow$ ) cho đỉnh đích:  $z(\downarrow, \phi, \infty)$

Tạo lập tập T gồm các đỉnh đã có nhãn lùi nhưng chưa được dùng để sinh nhãn lùi:  $T := \{z\}$ ; khởi gán điều kiện tăng luồng:  $\text{inc\_flow} := \text{False}$ ;

**Bước 2: Sinh nhãn**

**P1 Sinh nhãn tiến**

Trường hợp  $\text{Stop} = \text{True}$ ; thì xuất luồng cực đại, kết thúc.

Trường hợp  $\text{inc\_flow} = \text{True}$ ; chuyển sang thực hiện **bước 3**

2.1. Chọn đỉnh sinh nhãn tiến

\* Trường hợp  $S \neq \phi$ : Chọn đỉnh  $u \in S$  nhỏ nhất (theo thứ tự). Loại  $u$  khỏi S,  $S := S \setminus \{u\}$ . Ký hiệu nhãn tiến của  $u$  là  $(\uparrow, p, \alpha)$  và A là tập các đỉnh chưa có nhãn tiến kề với đỉnh  $u$ , Sang **bước 2.2**.

\* Trường hợp  $S = \phi$ , thì gán  $\text{Stop} := \text{True}$ ; thông báo hệ thống biết đã gặp điều kiện dừng, xuất luồng cực đại, kết thúc.

2.2. Gán nhãn tiến cho đỉnh chưa có nhãn tiến và kề đỉnh sinh nhãn tiến  $u$ .

Trường hợp  $\text{Stop} = \text{True}$ ; thì xuất luồng cực đại, kết thúc.

Trường hợp  $\text{inc\_flow} = \text{True}$ ; chuyển sang thực hiện **bước 3**

\* Trường hợp  $A = \phi$ : Quay lại **bước 2**.

\* Trường hợp  $A \neq \phi$ : Chọn  $t \in A$  nhỏ nhất (theo thứ tự). Loại  $t$  khỏi A,  $A := A \setminus \{t\}$ .

Gán nhãn tiến cho  $t$  như sau:

Nếu  $(u, t) \in E$  và  $f_{ut} < c_{ut}$ , đặt nhãn tiến đỉnh  $t$  là  $(\uparrow, u, \min\{\alpha, c_{u,t} - f_{u,t}\})$ .

Nếu  $(t, u) \in E$  và  $f_{tu} > 0$ , đặt nhãn tiến đỉnh  $t$  là  $(\uparrow, u, \min\{\alpha, f_{tu}\})$ .

Nếu  $t$  không được gán nhãn tiến, thì quay lại **bước 2.2**.

Nếu  $t$  được gán nhãn tiến và  $t$  có nhãn lùi, thì gán  $\text{inc\_flow} := \text{True}$ ; thông báo cho hệ thống biết đã tìm được đường đi tăng luồng, sang bước **3**, hiệu chỉnh tăng luồng, xóa nhãn.

Nếu  $t$  được gán nhãn tiến và  $t$  không gán nhãn lùi, thì bổ sung  $t$  vào S,  $S := S \cup \{t\}$  và quay ngược lại **bước 2.2**.

**P2: Sinh nhãn lùi**

Trường hợp  $\text{Stop} = \text{True}$ ; thì kết thúc.

Trường hợp  $\text{inc\_flow} = \text{True}$ ; chuyển sang thực hiện **bước 3**

2.3. Chọn đỉnh sinh nhãn lùi

\* Trường hợp  $T \neq \emptyset$ : Chọn đỉnh  $v \in T$  nhỏ nhất (theo thứ tự). Loại  $v$  khỏi  $T$ ,  $T := T \setminus \{v\}$ .

Ký hiệu nhãn lùi của  $v$  là  $(\downarrow, q, \beta)$  và  $B$  là tập các đỉnh chưa có nhãn lùi kề đỉnh sinh nhãn lùi  $v$ .

Sang **bước 2.4**.

\* Trường hợp  $T = \emptyset$ , thì **Stop = True**; thông báo hệ thống biết P2 đã gặp điều kiện dừng, kết thúc.

2.4. Gán nhãn lùi cho đỉnh chưa có nhãn lùi và kề đỉnh sinh nhãn lùi  $v$ .

Trường hợp **Stop = True**; thì kết thúc.

Trường hợp **inc\_flow = True**; chuyển sang thực hiện **bước 3**

\* Trường hợp  $B = \emptyset$ : Quay lại **bước 2**.

\* Trường hợp  $B \neq \emptyset$ : Chọn  $t \in B$  nhỏ nhất (theo thứ tự). Loại  $t$  khỏi  $B$ .  $B := B \setminus \{t\}$  gán nhãn lùi cho  $t$  như sau:

Nếu  $(t, v) \in E$  và  $f_{tv} < c_{tv}$ , đặt nhãn lùi đỉnh  $t$  là  $(\downarrow, v, \min\{\beta, c_{tv} - f_{tv}\})$ .

Nếu  $(v, t) \in E$  và  $f_{vt} > 0$ , đặt nhãn lùi đỉnh  $t$  là  $(\downarrow, v, \min\{\beta, f_{vt}\})$ .

Nếu  $t$  không được gán nhãn lùi, thì quay lại **bước 2.4**.

Nếu  $t$  được gán nhãn lùi và  $t$  có nhãn tiến thì gán **inc\_flow = True**; thông báo cho hệ thống biết P2 đã tìm được đường đi tăng luồng, sang **bước 3**, hiệu chỉnh tăng luồng, xóa nhãn.

Nếu  $t$  được gán nhãn lùi và  $t$  không có nhãn tiến, thì bổ sung  $t$  vào  $T$ ,  $T := T \cup \{t\}$  và quay lại **bước 2.4**.

**Bước 3: Hiệu chỉnh tăng luồng, xóa nhãn**

Ta có  $t$  là đỉnh được gán nhãn tiến ở bước 2.2 hoặc nhãn lùi ở bước 2.4 để thuật toán dẫn đến bước 3. Đỉnh  $t$  có nhãn tiến  $(\uparrow, p, \alpha)$  và nhãn lùi  $(\downarrow, q, \beta)$ . Đặt  $\delta = \min\{\alpha, \beta\}$ .

Ta hiệu chỉnh luồng  $f$  và xóa nhãn như sau.

### P1

3.1. Hiệu chỉnh ngược từ  $t$  về  $a$  theo nhãn tiến

3.1.1. Khởi tạo  $j := t, i := p$

3.1.2. Hiệu chỉnh

Nếu cung  $(i, j) \in G$ , thì hiệu chỉnh  $f_{ij} = f_{ij} + \delta$ .

Nếu cung  $(j, i) \in G$ , thì hiệu chỉnh  $f_{ij} = f_{ij} - \delta$ .

3.1.3. Tịnh tiến

Nếu  $i = a$ , thì xóa tất cả các nhãn tiến trên mạng trừ đỉnh nguồn  $a$  và đỉnh đích  $z$ , thông báo hệ thống biết P1 đã thực hiện việc tăng luồng, xóa nhãn tiến xong, đợi P2 xóa nhãn xong, quay lại **bước 2**.

Nếu  $i \neq a$ , thì đặt  $j := i$  và  $i := h$ , với  $h$  là thành phần thứ hai của nhãn tiến đỉnh  $j$ . Sau đó quay lại **bước 3.1.2**.

### P2

3.2. Hiệu chỉnh từ  $t$  đến  $z$  theo nhãn lùi

3.2.1. Khởi tạo  $i := t, j := q$

3.2.2. Hiệu chỉnh

Nếu cung  $(i, j) \in G$ , thì hiệu chỉnh  $f_{ij} = f_{ij} + \delta$ .

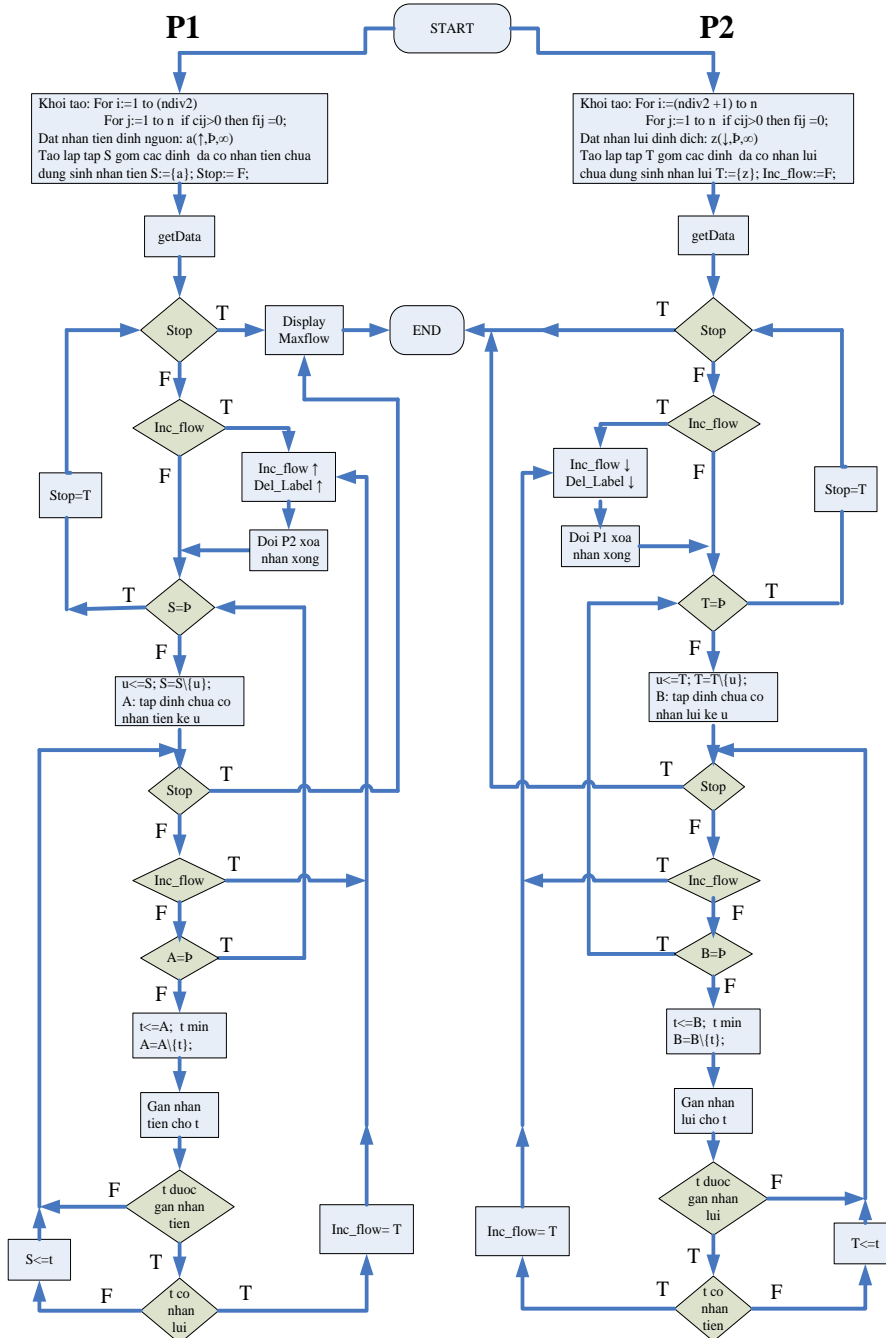
Nếu cung  $(j, i) \in G$ , thì hiệu chỉnh  $f_{ij} = f_{ij} - \delta$ .

### 3.2.3. Tinh tiến

Nếu  $i = z$ , thì xóa tất cả các nhãn lùi trên mạng trừ đỉnh nguồn a và đỉnh đích z, thông báo hệ thống biết P2 đã thực hiện việc tăng luồng, xoá nhãn lùi xong, đợi P1 xoá nhãn xong, quay lại **bước 2**.

Nếu  $i \neq z$ , thì đặt  $i := j$  và  $j := k$ , với k là thành phần thứ hai của nhãn lùi đỉnh i. Sau đó quay lại **bước 3.2.2**.

Sơ đồ mô tả thuật toán cho ở hình 1.



Hình 1: Sơ đồ thuật toán song song tìm luồng cực đại trên mạng

### **Định lý 1**

Nếu các khả năng thông qua có giá trị nguyên thì sau hữu hạn bước quá trình giải kết thúc.

#### **Chứng minh**

Qua mỗi bước hiệu chỉnh, giá trị luồng tăng lên ít nhất một đơn vị (do khả năng thông qua nguyên kéo theo  $\alpha$  và  $\beta$  nguyên dương nên  $\delta = \min\{\alpha, \beta\}$  cũng nguyên dương). Mặt khác giá trị luồng bị chặn bởi tổng khả năng thông qua của các cung đi ra khỏi đỉnh nguồn. Vì vậy qua một số hữu hạn bước quá trình giải kết thúc.

#### **Hệ quả**

Nếu các khả năng thông qua có giá trị hữu tỉ thì sau hữu hạn bước quá trình giải kết thúc.

#### **Chứng minh**

Qui đồng mẫu số các giá trị thông qua, giả sử mẫu số chung là  $N$ , qua mỗi bước tăng luồng, giá trị luồng tăng lên ít nhất  $1/N$ , mặt khác giá trị luồng bị chặn bởi tổng khả năng thông qua của tất cả các cung đi ra từ đỉnh nguồn. Vì vậy qua một số hữu hạn bước, quá trình giải kết thúc.

### **Định lý 2**

Cho mạng  $G(V, E, C)$ , nguồn  $a$ , đích  $z$ ,  $f$  là luồng nhận được sau khi kết thúc thuật toán, khi đó  $f$  là luồng cực đại.

#### **Chứng minh**

Ta xét hai trường hợp kết thúc thuật toán.

TH1: Thuật toán kết thúc xuất phát từ P1 gặp điều kiện  $S = \Phi$ . Kí hiệu  $X$  là tập tất cả các đỉnh mang nhãn tiến, khi đó lát cắt  $(X, V \setminus X)$  là lát cắt cực tiểu, suy ra  $f$  là luồng cực đại.

TH2: Thuật toán kết thúc xuất phát từ P2 gặp điều kiện  $T = \Phi$ . Kí hiệu  $Y$  là tập tất cả các đỉnh mang nhãn lùi, khi đó lát cắt  $(V \setminus Y, Y)$  là lát cắt cực tiểu, suy ra  $f$  là luồng cực đại.

## **5. Nhận xét**

➤ Việc cài đặt thuật toán song song là một vấn đề khó, tùy theo từng mô hình xử lý mà ta có cách cài đặt khác nhau, cần lưu ý một số vấn đề phát sinh khi cài đặt: Sau bước tăng luồng + xóa nhãn, việc Vi xử lý này chờ cho Vi xử lý kia thực hiện xóa nhãn xong rồi mới chuyển sang bước thực hiện gán nhãn (nếu không như vậy thì trong bước gán nhãn tiếp theo có thể rơi vào tình trạng gặp nhãn cũ chưa kịp xóa) có thể làm giảm tốc độ xử lý bài toán, đòi hỏi cần có kỹ thuật cài đặt tốt tránh trường hợp lãng phí thời gian xử lý. Trong bước khởi tạo, nếu hai vi xử lý tiến hành khởi tạo không cân sức sẽ dẫn đến tình trạng chưa nạp kịp dữ liệu mà đã xử lý, do đó trong quá trình cài đặt thuật toán trên các mô hình cụ thể, chúng ta phải giải quyết vấn đề này một cách hợp lý (đảm bảo trình tự các tiến trình và thời gian thực hiện). Trong trường hợp xấu nhất, hai vi xử lý cùng một lúc gặp đỉnh trung gian  $t$  và thực hiện gán nhãn cho  $t$  (cả nhãn tiến và nhãn lùi) tạo ra tình huống gặp điều kiện tăng luồng nhưng không được phát hiện. Nên khi cài đặt ta phải đảm bảo vấn đề đồng bộ hóa các tiến trình tác động trên cùng một đối tượng.

➤ Đây là thuật toán song song tổng quát, có thể áp dụng hiệu quả cho các loại mạng. Qua thực nghiệm chạy chương trình cho thấy sự phân chia công việc xử lý cho mỗi Vi xử lý gần như đều nhau (kết quả thực nghiệm cho thấy nút trung gian  $t$  tìm được nằm phân giữa 2 nút nguồn và đích trên đường tăng luồng). Nếu thuật toán được cài đặt trong môi trường xử lý song song tối ưu, dự kiến thời gian tính toán sẽ giảm đi một nửa so với thuật toán truyền thống.

## TÀI LIỆU THAM KHẢO

- [1] Trần Quốc Chiến, *Thuật toán hoán chuyển nguồn đích tìm luồng cực đại (1)*, Tạp chí Khoa học & Công nghệ, Đại học Đà Nẵng, 1(13)/2006, 53-58.
- [2] Trần Quốc Chiến (2006), *Thuật toán hoán chuyển nguồn đích tìm luồng cực đại (2)*, Tạp chí Khoa học & Công nghệ, Đại học Đà Nẵng, 3(15)-4(16)/2006, 77-82.
- [3] Trần Quốc Chiến (2005), *Giáo trình Lý thuyết đồ thị*, Đại học Đà Nẵng 2005.
- [4] Đoàn Văn Ban, Nguyễn Mậu Hân (2002), *Xử lý song song & phân tán*, Viện Công nghệ Thông tin.
- [5] Lê Minh Hoàng (2002), *Giải thuật & lập trình*, Đại học Sư phạm Hà Nội.
- [6] Nguyễn Tô Thành, Nguyễn Đức Nghĩa (1994), *Giáo trình Toán rời rạc*, Trường Đại học Bách khoa Hà Nội.
- [7] Ngọc Anh Thư (chủ biên - 2002), *Giáo trình Thuật toán*, Nhà xuất bản Thống kê.
- [8] L. R. Ford and D. R. Fulkerson (1962), *Flows in Networks*, Princeton University.
- [9] Reihard Diestel (2005), *Graphic Theory*, Electronic Edition, New York.