Equipment with new general education program, Volume 2, Issue 315 (June 2024)
ISSN 1859 - 0810

Ứng dụng giải thuật ngẫu nhiên để giải bài toán tìm đường đi ngắn nhất giữa các cặp đỉnh trong đồ thị

Huỳnh Huy Tuấn*

**Khoa Công nghệ thông tin, Trường Đại học Bạc Liêu*

Received: 10/5/2024; Accepted: 16/5/2024; Published: 27/5/2024

Abstract: *Graph theory is applied to solve real-life problems such as finding the shortest route between two cities in the traffic network, making timetables, distributing radio and television stations, etc. Common algorithms to solve this problem are: Dijkstra, Floyd-Warshall, Bellman-Ford, with approximately O(n3) complexity, where n is the number of vertices in the graph. When n is large enough, the complexity O(n3) is a significant number and the execution time is very high in practice. One solution to improve the complexity of the algorithm is to apply a random algorithm to solve the problem. In this article, I present the application of the Las Vegas random algorithm to solve with complexity lower than O(n3), with experimental results the complexity is equivalent to O(n2,376).*

Keywords: *Applied graph theory*

1. Giải thuật Las Vegas

Giải thuật Las Vegas là giải thuật ngẫu nhiên luôn cho kết quả chính xác, tuy nhiên thời gian chạy mỗi lần là khác nhau tùy thuộc vào đầu vào và được mong đợi trong một thời gian giới hạn nhất định. Các thuật toán Las Vegas nổi bật trong lĩnh vực trí tuệ nhân tạo và trong các lĩnh vực nghiên cứu hoạt động và khoa học máy tính khác.

Các thuật toán Las Vegas được giới thiệu vào năm 1979, Babai đã giới thiệu thuật ngữ "thuật toán Las Vegas" cùng với một ví dụ liên quan đến việc tung đồng xu.

Ví dụ: Tìm một số X trong dãy số có n số

```
Lặp lại các bước sau:
        k = RandInt(n)
        if A[k] == 1,
        return k;
```

Một biến chỉ số mảng k được tạo ngẫu nhiên, nếu phần tử A[k] = 1 thì k được trả về. Thuật toán có số lần lặp cho mỗi lần chạy là khác nhau do tính ngẫu nhiên của hàm RandInt(n).

2. Giải bài toán tìm đường đi ngắn nhất giữa các đỉnh của đồ thị bằng giải thuật Las Vegas

2.1. Mô tả bài toán

Cho $G(V,E)$ là một đồ thị vô hướng liên thông với tập đỉnh $V = \{1,\ldots,n\}$ và tập cạnh E với $|E| = m$. Ma trận kề A biểu diễn G là ma trận (0-1) cấp $n \times n$ với $A_{ij} = A_{ji} = 1$ nếu và chỉ nếu cạnh (i, j) thuộc E. Dựa vào A, chúng ta xác định ma trận khoảng cách D là ma trận số nguyên không âm cấp $n \times n$ sao cho D_{ij} là chiều dài đường đi ngắn nhất từ đỉnh i đến đỉnh j. Giá trị trên đường chéo của hai ma trận A và D là 0. Vì G liên thông nên tất cả các giá trị trong D được xác định.

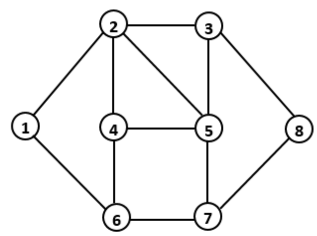
Ví dụ: Đồ thị G có 8 đỉnh:

***A* =**

	1	2	3	4	5	6	7	8
1	0	1	0	0	0	1	0	0
2	1	0	1	1	1	0	0	0
3	0	1	0	0	1	0	0	1
4	0	1	0	0	1	1	0	0
5	0	1	1	1	0	0	1	0
6	1	0	0	1	0	0	1	0
7	0	0	0	0	1	1	0	1
8	0	0	1	0	0	0	1	0

***D* =**

	1	2	3	4	5	6	7	8
1	0	1	2	2	2	1	2	3
2	1	0	1	1	1	2	2	2
3	2	1	0	2	1	3	2	1
4	2	1	2	0	1	1	2	3
5	2	1	1	1	0	2	1	2
6	1	2	3	1	2	0	1	2
7	2	2	2	2	1	1	0	1
8	3	2	1	3	2	2	1	0



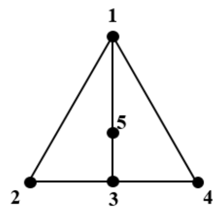
Hình 1: Đồ thị G

2.2. Xây dựng giải thuật

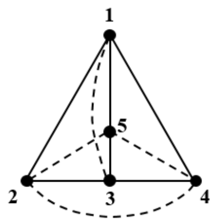
Bài toán tìm đường đi ngắn nhất giữa các cặp đỉnh (*The all-pairs shortest paths*)-Gọi tắt là Bài toán APSP có chiều dài đường đi được xác định trong ma trận khoảng cách D. Như vậy để giải Bài toán APSP trước tiên chúng ta sẽ tính đường đi ngắn nhất của từng cặp đỉnh trong đồ thị G-*the all-pairs distances*, gọi tắt là Bài toán APD.

2.3. Giải bài toán APD

Để giải bài toán APD, dùng các phép nhân ma trận số nguyên. Gọi $G'(V, E')$ là đồ thị có được từ $G(V,E)$ bằng cách xác định một cạnh của G' giữa cặp đỉnh $i \neq j \in V$ sao cho chúng có độ dài đường đi là 1 hoặc 2 trong G *(hình 1.1, 1.2)*. Do đó đồ thị G là đồ thị con của G' và chúng ta xem G' như là đồ thị "bình phương" của G. Từ G' ta biểu diễn A' là ma trận kề và D' là ma trận khoảng cách.



Hình 1.1 - Đồ thị G



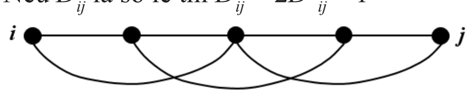
Hình 1.2 - Đồ thị G'

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad A' = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

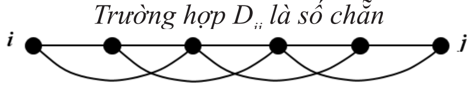
Gọi $Z = A^2$, trong đó A là ma trận kề của đồ thị G, khi đó có một đường đi giữa cặp đỉnh i và j chiều dài 2 trong G nếu và chỉ nếu $Z_{ij} > 0$.

Với mọi cặp đỉnh $i, j \in V$:

- Nếu D_{ij} là số chẵn thì $D_{ij} = 2D'_{ij}$
- Nếu D_{ij} là số lẻ thì $D_{ij} = 2D'_{ij} - 1$



Trường hợp D_{ij} là số chẵn



Trường hợp D_{ij} là số lẻ

Hình 2 – Dij là số chẵn và D_{ij} là số lẻ

Một ứng dụng trực tiếp là dựa vào D' của bài toán APD trong ma trận G' để tính ma trận D của bài toán APD trong ma trận G một cách nhanh chóng để tìm ra các đường ngắn nhất trong D. Từ đây chúng ta sẽ xét đến một giải thuật đệ qui cho bài toán APD như sau: trước tiên tính A' của G', dùng đệ qui để tính D', sau đó tính D từ D'. Với mọi cặp đỉnh xác định i và j trong G:

- Nếu D_{ij} là số chẵn thì $D'_{kj} \geq D'_{ij}$ với mọi lân cận k của i trong G
- Nếu D_{ij} là số lẻ thì $D'_{kj} \leq D'_{ij}$ với mọi lân cận k của i trong G. Ngoài ra sẽ tồn tại một lân cận k của i trong G sao cho $D'_{kj} < D'_{ij}$.

Giải thuật APD

Input: Đồ thị ***G(V,E)*** được biểu diễn bằng ma trận kề ***A***

Output : APD ma trận ***D*** cho đồ thị ***G***

1. $\boldsymbol{Z \leftarrow A^2}$
2. Tính ma trận ***A'*** sao cho $\boldsymbol{A'_{ij} = 1}$ nếu và chỉ nếu $i \neq j$ và ($\boldsymbol{A_{ij} = 1}$ hoặc $\boldsymbol{Z_{ij} > 0}$).
3. **If** $\boldsymbol{A'_{ij} = 1}$ với mọi $i \neq j$ **then Return** $\boldsymbol{D = 2A' - A}$.
4. Tính đệ qui cho **APD** ma trận ***D'*** cho đồ thị ***G'*** với ma trận kề ***A'***.
5. $\boldsymbol{S \leftarrow AD'}$.

Return ma trận ***D*** với $\boldsymbol{D_{ij}} = \begin{cases} 2D_j^{'} & f\ S_j \geq D_j^{'}.d(i) \\ 2D_j^{'} - 1 & f\ S_j < D_j^{'}.d(i) \end{cases}$

**** Cách tính D' bằng đệ qui Ở bước 4 trong giải thuật APD***

- Trước tiên ta khởi tạo cho ma trận D' bằng cách đặt: $D'_{ij} = 0$ với $i = j$; $D'_{ij} = \infty$ với $i \neq j$

- Cho $D'_{ij} = A'_{ij}$ với mọi i, j thỏa $A'_{ij} = 1$, ngược lại tính đệ qui bằng cách tính: $A' = A' \times A'$ cho đến khi $A'_{ij} = 1$ với mọi i, j.

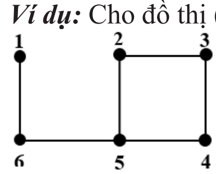
- Nếu $A'_{ij} = 1$ thì $D'_{ij} = 2k$ (k là số lần lặp của giải thuật APD) với mọi i, j thỏa $D'_{ij} = \infty$.

Giải thuật APD tính khoảng cách ma trận cho đồ thị G có n đỉnh trong thời gian $O(MM(n)\log n)$ trong đó sử dụng phép nhân các ma trận số nguyên với các giá trị vào được giới hạn bởi $O(n^2)$.

2.3. Nhân ma trận logic ghi nhận đường đi (witness)

Dùng phương pháp ngẫu nhiên để giải bài toán APSP là tìm ma trận ghi nhận đường đi (witness) cho phép nhân ma trận logic. Cho A và B là hai ma trận logic cấp $n \times n$, $P = AB$ là ma trận tích. Một vị trí ghi nhận cho P_{ij} là một chỉ số $k \in \{1,...,n\}$ sao cho $A_{ik} = B_{kj} = 1$. Biết rằng $P_{ij} = 1$ nếu và chỉ nếu tồn tại chỉ số k. Một ma trận logic ghi nhận đường đi (*Boolean product witness matrix)*-BPWM cho P là một ma trận số nguyên W sao cho với mỗi W_{ij} chứa một chỉ số k cho P_{ij} nếu tồn tại và $P_{ij} = 0$ nếu không tồn tại. Ma trận W có giá trị nằm trong tập $\{0, 1, ..., n\}$. Bài toán BPWM là tìm một ma trận ghi nhận đường đi W.

Ví dụ: Cho đồ thị (hình 3)



Hình 3

Ma trận kề biểu diễn cho đồ thị là A và B

A =

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	0
5	0	1	0	1	0	1
6	1	0	0	0	1	0

B =

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	0
5	0	1	0	1	0	1
6	1	0	0	0	1	0

ISSN 1859 - 0810

Suy ra:

$C =$

	1	2	3	4	5	6
1	1	0	0	0	1	0
2	0	2	0	2	0	1
3	0	0	2	0	2	0
4	0	2	0	2	0	1
5	1	0	2	0	3	0
6	0	1	0	1	0	2

$P =$

	1	2	3	4	5	6
1	1	0	0	0	1	0
2	0	1	0	1	0	1
3	0	0	1	0	1	0
4	0	1	0	1	0	1
5	1	0	1	0	1	0
6	0	1	0	1	0	1

Ma trân ghi nhân đường đi cho *P* là

$W =$

	1	2	3	4	5	6
1	6	0	0	0	6	0
2	0	3	0	3	0	5
3	0	0	4	0	4	0
4	0	3	0	3	0	5
5	6	0	4	0	4	0
6	0	5	0	5	0	1

Giải thuật BPWM

Input: A và B là hai ma trận logic (0-1) cấp $n \times n$
Output: Ma trận "witness" W cho ma trận logic $P = AB$.
1. $W \leftarrow -AB$
2. **For** $t = 0$ to $\log n$ **do**
2.1 $r \leftarrow 2^t$;
2.2 **Repeat** $\lceil 3.77 \log n \rceil$ **lần**
2.2.1 Chọn số ngẫu nhiên $R \subseteq \{1, . . . , n\}$ với $|R| = r$
2.2.2 Tính A^R và B^R;
2.2.3 $Z \leftarrow A^R B^R$;
2.2.4 **For all** (i,j) **do**
If $W_{ij} < 0$ and Z_{ij} là "witness" **then** $W_{ij} \leftarrow Z_{ij}$;
3. **For all** (i,j) **do**
If $W_{ij} < 0$ **then**
Tìm "witness" W_{ij} bằng phương pháp brute-force (liệt kê).

Giải thuật BPWM là một giải thuật Las Vegas cho bài toán BPWM với hy vọng chạy trong thời gian O(MM(n)log$^2 n$), *Trong đó O(MM(n)) là thời gian chạy cho giải thuật nhân 2 ma trận n × n* ≈ O($n^{2.376}$)

2.4. Xác định đường ngắn nhất

Cuối cùng, giải bài toán APSP sẽ thông qua giải thuật APD và BPWM. Bài toán APD cần phải tồn tại một đồ thị với nhiều cặp đỉnh mà độ dài của đường ngắn nhất tuyến tính với *n*.

Giải thuật APSP

Input: Ma trận kề A cấp n × n của đồ thị G
Output: Ma trận Kết quả "*Successor*" S cho đồ thị G
1. Tính ma trận khoảng cách $D = APD(A)$
2. **For** $s = \{0, 1, 2\}$ **do**
2.1 Tính ma trận (0-1) $D^{(s)}$ sao cho $D_{kj}^{(s)} = 1$ nếu và chỉ nếu $D_{kj} + 1 \equiv s \bmod 3$
2.2 Tính ma trận $W^{(s)} = BPWM(A,D^{(s)})$.
3. Tính ma trận Kết quả *Successor* S sao cho

$$S_{ij} = W_{ij}^{(D_{ij} \bmod 3)}$$

Với đồ thị như (hình 3), ta có ma trận kề A, sau khi thi hành giải thuật APSP ta sẽ thu ma trận khoảng cách *D* và ma trận Kết quả S như sau:

$A =$

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	0
5	0	1	0	1	0	1
6	1	0	0	0	1	0

$D =$

	1	2	3	4	5	6
1	0	3	4	3	2	1
2	3	0	1	2	1	2
3	4	1	0	1	2	3
4	3	2	1	0	1	2
5	2	1	2	1	0	1
6	1	2	3	2	1	0

$S =$

	1	2	3	4	5	6
1	0	6	6	6	6	6
2	5	0	3	5	5	5
3	2	2	0	4	2	2
4	5	5	3	0	5	5
5	6	2	2	4	0	6
6	1	5	5	5	5	0

Giải thuật APSP tính ma trận "Kết quả" cho đồ thị G có *n* đỉnh trong thời gian mong đợi là: O(MM(n) log$^2 n$).

Chứng minh:

- Ở bước 1: Tính ma trận *D* bằng giải thuật APD có độ phức tạp là O(MM(n)log$n$).

- Ở bước 2:

+ Tính ma trận $D^{(S)}$ có độ phức tạp O(n^2).

+ Tính $W^{(s)}$ = BPWM(A,$D^{(s)}$) có độ phức tạp O(MM(n)log$^2 n$)

- Ở bước 3: Tính ma trận Kết quả *S* với $S_{ij} = W_{ij}^{(D_{ij} \bmod 3)}$ có độ phức tạp O(n^2).

Như vậy độ phức tạp cho giải thuật APSP là:

O(MM(n)log n) + O(n^2) + O(MM(n)log$^2 n$) + O(n^2) = O(MM(n)log$^2 n$) = O($n^{2.376}$ log$^2 n$)

3. So sánh và kết luận

3.1. So sánh các giải thuật tìm đường đi ngắn nhất

TT	Giải thuật	Độ phức tạp	Ghi chú
1	Dijkstra	O(n^3)	
2	Floyd-Warshall	O(n^3)	
3	APSP	O($n^{2.376}$ log$^2 n$)	

3.2. Kết luận

Bài toán tìm đường đi ngắn nhất giữa các cặp đỉnh trong đồ thị dùng giải thuật Las Vegas luôn cho kết quả đúng với thời gian chạy được mong đợi là O($n^{2.376}$ log$^2 n$). Đối với đồ thị nhiều đỉnh, cạnh thì giải thuật APSP sẽ cho thời gian chạy nhanh, hiệu quả hơn giải thuật Dijsktra, Floyd.

Tài liệu tham khảo

[1]. Rajeev Motwani & Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press, 2000.

[2]. Reinhard Diestel, Graph Theory, Springer-Verlay New Your 1997, 2000.

[3]. J.A.Bondy and U.S.R.Murty, *Graph Theory With Applications, American Elsevier,* New York, 1977.

[4]. Kenneth H. Rosen, *Toán học rời rạc ứng dụng trong tin học, bản dịch tiếng Việt,* NXB Khoa học và kỹ thuật, 2000.