THUẬT TOÁN GIA LƯỢNG NGẪU NHIÊN TÌM GIAO CỦA N NỬA - KHÔNG GIAN TRONG KHÔNG GIAN BA CHIỀU

Randomized incremental algorithm Searching for the intersection of n half-space in three-dimension space

Nguyễn Khắc Quốc[1]

Tóm tắt

Trong hình học, các đối tượng như điểm, đường thẳng, đa giác là cơ sở của các bài toán và các thuật toán. Một số thuật toán hình học dùng để phân tích, thiết kế các đối tượng vật lý và một số ứng dụng khác trong toán thống kê - các lĩnh vực mà trong đó nhiều vấn đề có thể giải quyết một cách đơn giản bằng đối tượng hình học. Nhưng việc xây dựng thuật toán để giải các bài toán hình học với độ chính xác cao, độ phức tạp thấp và thời gian chạy nhanh là một vấn đề rất phức tạp. Thời gian thực hiện của một thuật toán hình học thường phụ thuộc vào sự phân bố, thứ tự của các điểm ở đầu vào việc sử dụng các hàm lượng giác. Mục tiêu của bài báo này là phát triển thuật toán gia lượng ngẫu nhiên tìm giao của n nửa - không gian trong không gian 3 chiều. Thuật toán này chạy với thời gian là O(n log n).

Từ khóa: ngẫu nhiên, thuật toán, nửa không gian, ba chiều, hình học.

Abstract

In geometry, the objects as points, lines, polygons are the basis of the problems and algorithms. Some geometric algorithms are used to analyze, design physical objects and other applications in statistical mathematics in which many problems can simply be solved by geometric object. However, the construction of algorithms to solve geometric problems with high precision, low complexity and fast-run time is a very complex issue. The implementing time of a geometric algorithm depend much more on the distribution of points, their orders at the input and the use of the trigonometric functions. The objective of this paper is to develop a randomized incremental algorithm in order to seek for the intersection of n half-space in three-dimension space. This algorithm is carried out with the time of O(n log n).

Keywords: randomized, algorithm, half-space, three-dimension, geometry.

1. Giới thiệu

Đối tượng nghiên cứu cơ bản trong hình học là điểm và đoạn thẳng, những đối tượng phức tạp hơn như mặt phẳng và không gian. Một số bài toán nổi bật như: bao lồi, tam giác phân Delaunay, đường kính của một tập hợp điểm, các lớp lồi của một tập hợp,... Điển hình là sự phát hiện của Chand và Kapur 1970 và Jarvis 1973 với thuật toán bọc gói, Ronald Graham 1972 với thuật toán quét Graham, W.Eddy 1977 và A. Bykatnăm 1978 với thuật toán Quickhull, Timothy Chan 1993 với thuật toán Chan. Những thuật toán này phần lớn giải quyết các vấn đề trong hình học phẳng. Đối với những bài toán trong không gian, các thuật toán chỉ dừng lại ở không gian ba chiều và số không gian như Shamos 1975, Shamos và Hoey 1976 mô tả một thuật toán với thời gian là O (n log n) với số chiều là hai, Zolnowsky 1977, Preparata và Muller 1977 đã phát triển thuật toán với thời gian là O (n log n) trong không gian ba chiều. Nguyễn Khắc Quốc 2014 dựa trên cơ sở nghiên cứu của Rejeev Motwani 2000, phát triển thuật toán quy hoạch tuyến tính dựa trên gia lượng ngẫu nhiên với thời gian O (n log n).

2. Nội dung

Mối quan hệ giữa khoảng cách, góc, mặt phẳng và không gian đã được nhà toán học Euclide, người Hy Lạp, nghiên cứu khoảng 300 năm TCN. Các quan hệ này được nhiều người biết đến với tên gọi là "Hình học Euclide" hai hoặc ba chiều.

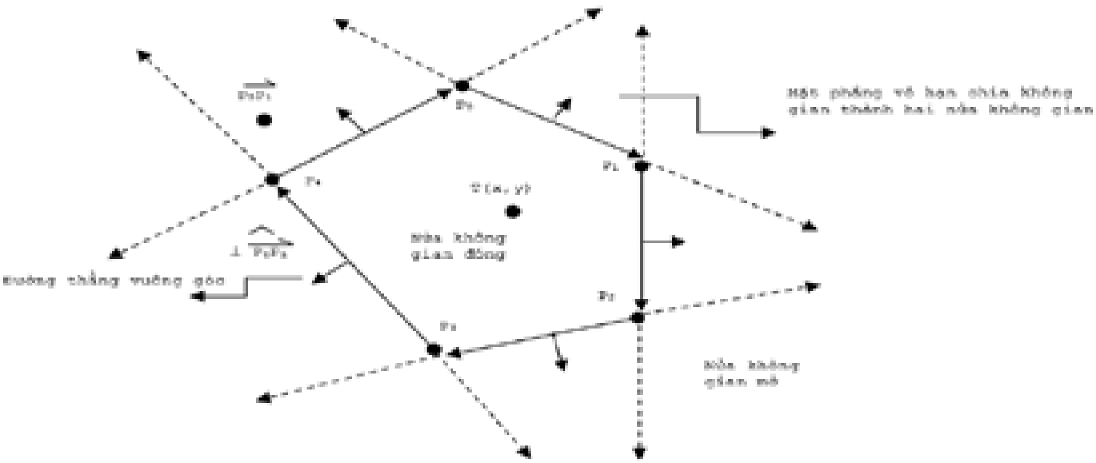
Trong toán học hiện đại, các quan hệ trên đã được tổng quát hóa cho các không gian bốn chiều, năm chiều và nhiều chiều hơn. Không gian *n-chiều* thỏa mãn các quan hệ Euclide được gọi là "không gian Euclide *n-chiều*".

2.1. Không gian và nửa không gian

- Trong toán học và vật lý học, không gian được xem là tập hợp những điều kiện để các sự vật và hiện tượng diễn ra.

- Trong hình học không gian, một nửa không gian là một trong hai phần của mặt phẳng chia không gian Euclide ba chiều. Một cách tổng quát hơn, một nửa không gian là một trong hai phần nào

[1] *Thạc sĩ, Bộ môn Công nghệ Thông tin Trường Đại học Trà Vinh*



Hình: Nửa không gian

đó của một siêu phẳng chia một không gian Affine. Đó là các điểm không để các siêu phẳng phân chia thành hai tập lồi đó là nửa không gian. Như vậy, bất kỳ không gian con nối một điểm trong tập điểm khác phải cắt nhau siêu phẳng.

2.2. Thuật toán gia lượng ngẫu nhiên tìm giao của n nửa không gian

Trước tiên, chúng ta xét thuật toán tìm bao lồi tuyến tính của *n* điểm trong không gian ba chiều với thời gian chạy là O(*n* log *n*).

Mô tả thuật toán

Cho tập *S* chứa *n* nửa - không gian trong không gian ba chiều, gọi *inter*(*S*) là tập chứa phần giao của chúng (*có thể rỗng*), tập này là một khối đa diện lồi trong không gian. Lưu ý rằng ở sự giao nhau này, chúng ta không quan tâm đến đường biên. Mọi mặt của đa diện này được chứa trong đường biên mặt phẳng của một trong nửa - không gian. Chúng ta cho rằng mỗi nửa - không gian như là một bất đẳng thức tuyến tính. Đường này có thể thay đổi hệ tọa độ; mỗi phép toán bằng nhau tương ứng được xác định bằng đường biên của mặt phẳng nằm trong nửa - không gian. Khi *inter*(*S*) là một đa giác (*không rỗng*), chúng ta có thể mô tả nó như là một đồ thị mà mỗi đỉnh của nó tương ứng với một đỉnh của đa diện, với các đỉnh liền kề của một đồ thị nếu các đỉnh tương ứng trong đa diện được kết hợp bởi một đường thẳng tạo thành trên bề mặt bằng sự giao nhau của hai nửa - không gian trong *S*. Khi *inter*(*S*) không có đường viền, điều này rất thuận lợi để tìm một điểm ở "*vô cực*", điểm này là điểm cuối của tất cả các cạnh nằm ở *nửa – vô cực* của đa diện. Như vậy, từ tập *S,* kết quả chúng ta cần tìm là một đồ thị mô tả các mặt của đa diện *inter*(*S*); chúng ta mô tả đồ thị này được tạo bởi một vị trí (*trong không gian*) của tất cả các đỉnh, cùng với sự liền kề giữa các đỉnh này.

Ngay khi mọi mặt của đa diện này nằm trong đường biên mặt phẳng của một trong các nửa - không gian và không có mặt phẳng nào chứa hơn một mặt của đa diện. Số các mặt nhiều nhất của đa diện là *n*. Vì vậy, đồ thị biểu diễn *inter*(*S*) là một đồ thị phẳng, trong đó số các đỉnh và các cạnh cả hai được tính là O(*n*). Chúng ta cho rằng sẽ không có bốn đường biên mặt phẳng cắt ngang qua một điểm chung. Vì thế, mọi đỉnh của đa diện là một đồ thị bậc ba (*khi cần thiết chúng ta có thể chấp nhận các đỉnh ở vô cực*). Khi đó chúng ta nói các cạnh liền kề với một đỉnh hay các mặt của đa diện tương ứng với các mặt của đồ thị liền kề với các đỉnh; thật vậy có ba mặt liền kề với mỗi đỉnh của *inter*(*S*). Tương tự, chúng ta cũng có thể chứng minh cho các cạnh biên của một mặt, và của hai mặt bên khác của các cạnh còn lại.

Thuật toán **(Half-Space Intersect Algorithm)**

Input: Tập S chứa *n* nửa - không gian

Output: *inter*(*S*) {tập chứa phần giao của *n* nửa - không gian}

0. Với $1 \le i \le n$

1.Thêm ngẫu nhiên nữa - không gian h_i *vào inter*(S_{i-1})

1.1. Loại bỏ tất cả các đỉnh trong $v \in \overline{h}_i \cap inter(S_{i-1})$;

1.2. Thêm đỉnh mới vào $h_i inter(S_{i-1})$; *{tại bước thứ i}*

2. Dùng con trỏ 2 chiều từ $h \in S \setminus S_i$ đến $v \in \overline{h}_i \cap inter(S_{i-1})$;

3. Thêm h_i vào đỉnh đầu tiên của đa giác;

4. Triệt tiêu các đỉnh cũ, tạo ra đỉnh mới trong $h_i \cap inter(S_{i-1})$;

5. Lặp lại 1 → 4;

Phân tích thuật toán

Thuật toán ngẫu nhiên tìm *inter*(S) là rất đơn giản. Chúng ta mô tả nó giống như việc tìm bao lồi tuyến tính của các điểm trong một mặt phẳng. Trước tiên, thuật toán thay đổi thứ tự ngẫu nhiên đầu vào của tập (S) chứa các nửa - không gian; gọi h_i là một nửa - không gian thứ i trong thứ tự ngẫu nhiên này; cho $1 \leq i$ n. Gọi S_i chứa tập $\{h_1 ... h_i\}$. Kế tiếp, thuật toán tiếp tục thực hiện qua n bước. Sau bước thứ i, thuật toán sẽ tìm được *inter*(S_i). Ngay ở bước thứ i, h_i được thêm vào *inter*(S_{i-1}), *inter*(S_i) là một tập trung gian trong tiến trình. Trong hình học, chúng ta có thể nhìn thấy như nó được cắt ra từ một phần của *inter* (S_{i-1}) không chứa h_i. Trong tiến trình này, một vài đỉnh của đa diện *inter*(S_{i-1}) đã bị xóa và một vài đỉnh mới được thêm vào. Tiếp theo, chúng ta sẽ mô tả chi tiết và sau đó sẽ phân tích tiến trình này.

Đầu tiên chúng ta cho rằng, điểm giao nhau của $\{h_1, h_2, h_3, h_4\}$ là đã được bao quanh. Thật vậy, *inter*(S_i) sẽ là một đa diện được bao quanh bởi một đường biên. Điều này chúng ta sẽ được nhận thấy trong suốt cả thuật toán.

Gọi $S\backslash S_i$ là tập chứa các nửa - không gian chưa được thêm vào sau bước thứ i. Chúng ta mô tả chỉ với nửa - không gian trong tập $S\backslash S_i$, nửa - không gian mà mặt phẳng giao *inter*(S_{i-1}) bao quanh; rõ ràng nó sẽ trở nên dễ dàng giải quyết khi còn lại nửa - không gian. Cho nhiều nửa - không gian h, gọi $\bar{h}$ là phần bù của h. Cho một nửa - không gian h, chúng ta nói rằng đỉnh của *inte*(S_{i-1}) *đối lập* với h nếu nó nằm trong $\bar{h}$. Với mỗi nửa - không gian $h \in S\backslash S_i$, chúng ta sử dụng con trỏ (*hai chiều*) trỏ đến các đỉnh của *inter*(S_{i-1}) đối lập với h. Như vậy, giả thiết bên dưới của thuật toán là hoàn toàn đúng đắn.

Quá trình thêm h_i vào *inter*(S_i) bắt đầu tại một đỉnh của *inter*(S_{i-1}) đối lập với h_i. Bắt đầu tại đỉnh này, chúng ta tìm cách để mô tả đồ thị *inter*(S_{i-1}),chúng ta không "thêm vào" *inter*(S_{i-1}) $\cap$ h_i. Trong tiến trình tìm kiếm này chúng ta xác định các đỉnh và các cạnh của *inter*(S_{i-1}) đã được triệt tiêu khi thêm vào h_i, và việc tạo thêm các đỉnh mới của *inter*(S_i) (*tất cả những đỉnh này nằm trong đường biên của mặt phẳng* h_i).

Rõ ràng, đánh giá việc tìm kiếm này là tỷ lệ với tổng số các đỉnh bị triệt tiêu và tổng số các đỉnh được tạo ra. Như chúng ta đã phân tích thuật toán tìm bao lồi tuyến tính trong không gian hai chiều, có lẽ chúng ta bỏ qua đánh giá việc xóa, ngay khi hầu hết các đỉnh đã được xóa đối lập với các đỉnh đã được tạo. Để phân tích số các đỉnh được tạo ra bởi việc thêm vào h_i, chúng ta cần phân tích ngược một lần nữa. Giả sử rằng, chúng ta có *inter*(S_i), khi chúng ta chọn xóa ngẫu nhiên nửa - không gian. Như vậy, khi chúng ta dùng cách này thì số các cạnh và số các đỉnh trong đồ thị phẳng với k mặt là O(k).

Nó vẫn còn đúng với mỗi nửa - không gian $h \in S\backslash S_i$, chúng ta dùng con trỏ (*hai chiều*) trỏ đến một đỉnh của *inter*(S_{i-1}) đối lập với h. Bây giờ, chúng ta làm như thế nào để có thể duy trì được con trỏ; và sau đó đánh giá việc làm này. Đặc biệt, chúng ta phải chỉ ra con trỏ thực hiện như thế nào cho nửa - không gian $S\backslash S_i$ và đã được cập nhật khi thêm vào h_i.

Đánh giá thuật toán

Khi triệt tiêu một đỉnh v của *inter*(S_{i-1}) ngay khi thêm vào h_i, chúng ta kiểm tra có hay không bất kỳ con trỏ từ v đến nửa - không gian $S\backslash S_i$. Với mỗi con trỏ (trỏ đến một nửa - không gian $h \in S\backslash S_i$), chúng ta phải dịch chuyển nó đến một đỉnh mới $w \in h \cap$ *inter*(S_i). Như vậy, chúng ta sẽ tìm đỉnh w như thế nào? Một tiến trình đơn giản là sử dụng việc cập nhật *inter*(S_{i-1}) vào một *inter*(S_i). Chú ý rằng, đỉnh v là nằm trong $\bar{h} \cap \bar{h}_i$. Chúng ta sẽ thực hiện một đường đi trong đồ thị biểu diễn bởi *inter*(S_{i-1}) bắt đầu tại v, cho đến khi chúng ta đến đỉnh đầu tiên của *inter*(S_i). Khi đi đến đỉnh đầu tiên của *inter*(S_i), chúng ta sẽ tìm thấy một đỉnh mới w cần tìm, nó nằm trong $\bar{h}$ và đối lập với h. Lúc này, chúng ta di chuyển con trỏ hai chiều từ vị trí đối lập với h trỏ đến w.

Chúng ta thực hiện với mọi đỉnh của đồ thị có bậc là ba. Tuy nhiên, đánh giá việc tìm kiếm này là tỷ lệ với số đỉnh của $\bar{h} \cap \bar{h}_i \cap$ *inter*(S_{i-1}), nó tương đương với số đỉnh bị triệt tiêu của *inter*(S_{i-1}) đối lập với h. Vì rằng, tổng đánh giá đường tiệm cận cho sự duy trì con trỏ h, nó chỉ thỏa mãn để đếm các đỉnh mới được tạo, khi bất kỳ đỉnh bị triệt tiêu đã được đếm lần nữa khi được tạo.

Như vậy, dự kiến cận của các đỉnh mới được tạo đối lập với h, tổng này sẽ nhiều hơn $h \in S\backslash S_i$. Điều này là chắc chắn.

$$\sum_{r} \left|\{h \in S \setminus S_i : h \text{ đối lập với } v\}\right| \qquad (1)$$

Tổng này có nhiều hơn một tập của các đỉnh trong *inter*(S_i) mới được tạo bằng việc thêm vào h_i. Chúng ta lấy cận của (1).

Tiếp theo, cho một tập của nửa - không gian H, gọi $c(H, h)$ chứa số các đỉnh của *inter*(H) đối lập với h. Việc sắp xếp được phân tích ngược lại lần nữa. Đầu tiên chúng ta xét tập S_i cố định từ việc xóa ngẫu nhiên nửa - không gian của $inter(S_{i-1})$. Không có đỉnh nào của $inter(S_i)$ có bậc là ba, dự kiến cận của (1) sẽ là cận của:

$$\frac{3}{i}\sum_{h\in S\setminus S_i} c(S_i, h) \qquad (2)$$

Ngay khi h_{i+1} đã được chọn ngẫu nhiên từ $S\setminus S_i$ ta có:

$$E[c(S_i, h_{i+1})] = \frac{1}{n-i}\sum_{h\in S\setminus S_i} c(S_i, h) \qquad (3)$$

Tổ hợp (2) và (3) , ta được cận của {1} bên dưới:

$$\frac{3(n-i)}{i} E[c(S_i, h_{i+1}]$$

Trong đó, biến ngẫu nhiên $c(S_i, h_{i+1})$ là biến đếm số đỉnh bị triệt tiêu bằng việc thêm vào h_{i+1}, hay nửa - không gian được thêm vào ở bước $i + 1$. Thật vậy, tổng tất cả i của (1) ta được cận bên dưới:

$$\sum_{i=1}^{n}\frac{3(n-1)}{i} E\,[\text{số đỉnh bị triệt tiêu tại thời điểm } i+1] \qquad (4)$$

Gọi v là đỉnh được tạo trong tiến trình của thuật toán, gọi $t_c(v)$ là tổng thời gian (số bước) tại thời điểm nó được tạo, và $t_d(v)$ là thời gian tại thời điểm nó bị triệt tiêu. Khi đó, (4) có thể được viết lại như sau:

$$\sum_{v}\frac{3(n-t_d(v)-1)}{t_d(v)-1} \qquad (5)$$

Ở đây v sẽ biến thiên theo các đỉnh khi được tạo và quá trình thực hiện của thuật toán. Khi $t_c(v) \leq t_d(v)$ -1, chúng ta có cận của (5) bên dưới:

$$\sum_{v}\frac{3(n-t_c(v)}{t_c(v)} \leq \sum_{i=1}^{n}\frac{3(n-1)}{i} E[\,|\{v|t_c(v)=i\}|\,]$$

Chúng ta dễ dàng nhận thấy rằng

E = $[|\{v \mid t_c(v) = i\}|]$ là một hằng số.

Như vậy, chúng ta có thể phát biểu định lý sau:

Định lý: *Thời gian chạy dự kiến của thuật toán gia lượng ngẫu nhiên để tìm giao của n nửa - không gian trong không gian 3 chiều là* **O*n*(*n* log *n*)**.

3. Kết luận

Trong toán học và tin học, những thuật toán hình học thường khó phân tích hơn các thuật toán trong những lĩnh vực khác. Các thuật toán hình học khó mô tả các dữ liệu đầu vào, đầu ra. Chính vì vậy, việc phân tích hay xây dựng một thuật toán hình học đòi hỏi chúng ta phải hiểu biết rõ về sự tác động qua lại rất phức tạp giữa các đặc tính của các tập điểm, các cạnh hay các không gian. Thuật toán này đã cải tiến được thời gian chạy cho thuật toán tất định tìm *n – nửa không gian*. Đồng thời, chúng ta có thể ứng dụng thuật toán này vào trong nhiều lĩnh vực khác.

Tài liệu tham khảo

Nguyễn, Khắc Quốc. 2014. "Xây dựng thuật toán quy hoạch tuyến tính dựa trên gia lượng ngẫu nhiên", *Tạp chí Khoa học*, Trường Đại học Trà Vinh, Số 14, tr.11-17.

Rejeev Motwani. 2000. *Randomized Algorithm.* Stanford University, tr. 241-244.

Shamos, M.I. 1975. "Geometric Complexity". *Proc. Seventh Annual AC M Symposium on Automata and Computability Theory*, tr. 224-233.

Shamos, M.I, and Hoey, D. 1975. "Closest-Point problems". *Proc. Sixteenth Annual IEEE Symposium on Foundations of Computing*, October, tr. 151-162.

Shamos, M.I, and Hoey, D. 1976. "Geometric Intersection Problems". *Proc. Seventeenth Annual IEEE Symposium on Foudations of Computer Science*, October, tr. 208-215.