

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Nhập môn phân tích độ phức tạp thuật toán
Báo cáo sơ bộ bài tìm hiểu P vs NP

GVLT: Thầy Trần Đan Thư

GVTH: Thầy Nguyễn Đức Thân
Thầy Trương Toàn Thịnh
Thầy Nguyễn Vinh Tiệp
Thầy Nguyễn Sơn Hoàng Quốc

Nhóm: 13

Sv:	Nguyễn Phan Mạnh Hùng	1312727
	Lục Kiến Nghiệp	1312734
	La Ngọc Thùy An	1312716
	Nguyễn Phước Đạt	1312721

1 Tổng quan

1.1 Phát biểu bài toán

Bài toán được phát biểu như sau:

Liệu tất cả các bài toán mà lời giải của nó có thể kiểm tra được trong thời gian đa thức có thể giải trong thời gian đa thức được hay không?

1.2 Sơ lược về nguồn gốc bài toán

Bài toán được đề cập tới lần đầu trong một bức thư do Kurt Godel gửi tới John Von Neumann vào năm 1956. Trong đó Godel đặt vấn đề liệu các bài toán NP có thể giải trong thời gian tuyến tính hay bậc hai được không. Tuy vậy bài toán này được phát biểu một cách chính xác trong bài báo của Stephen Cook năm 1971, tên "The complexity of theorem proving procedures". Tuy bài toán này còn khá "non trẻ" nếu đặt lên bàn cân so với nhiều bài toán nổi tiếng khác, nhưng nó được xem là một trong những vấn đề quan trọng nhất lĩnh vực khoa học máy tính và toán học, và còn được xem như một cầu nối giữa hai lĩnh vực này. Chính bởi tính quan trọng của nó mà việc toán học Clay (Clay Mathematics Institute) đã lựa chọn nó là một trong 7 bài toán triệu đô.

1.3 Tầm quan trọng của bài toán

Để có thể hiểu rõ được tầm quan trọng của việc giải bài toán này ta sẽ thử nhìn từ 2 góc độ sau.

Điều gì sẽ xảy ra nếu:

- $P = NP$
- $P \neq NP$

1.3.1 $P = NP$

Nếu $P = NP$ và, quan trọng hơn nữa, ta tìm được một cách giải hiệu quả cho một bài toán thuộc lớp NP-Complete. Khi đó, có thể nói, thế giới hoàn toàn thay đổi bởi khi ấy các ta có thể giải được các bài toán tối ưu một cách hiệu quả. Khi đó

- Ta có thể tạo ra các hệ thống lập lịch hoàn hảo để tiết kiệm tối đa nguồn lực, nâng cao hiệu suất, giảm thiểu lãng phí trong các bài toán di chuyển, xếp lịch làm việc.
- Ta có thể giải quyết nhiều vấn đề nan giải trong y học và chữa trị được nhiều bệnh nan y.
- Nâng cao khả năng học tập, học ít nhất nhưng đạt hiệu quả cao nhất.
- Và ta cũng có thể giải quyết hầu hết mọi bí ẩn mà khoa học chưa tìm ra đáp án.

Tuy vậy, bên cạnh những lợi ích mang lại, thì nhân loại cũng phải đối mặt với nhiều vấn đề mà nó mang lại, đặc biệt là vấn đề an ninh. Bởi nhiều hệ thống an ninh được thiết lập dựa trên nền tảng của những bài toán khó, không chỉ là trong trường hợp xấu nhất mà còn phải đảm bảo các bài toán nền tảng này khó giải được trong các trường hợp trung bình. Nói rõ hơn, khi đó:

- Các hệ thống sử dụng mã công khai, public key, sẽ sụp đổ.
- Các hệ thống tin nhiệm (trusted third party, trusted servers) sẽ trở nên mất an toàn.

1.3.2 $P \neq NP$

Khi $P \neq NP$, chúng ta sẽ phải chấp nhận thế giới đầy những vấn đề không giải quyết được. Tuy vậy, con người cũng sẽ định hướng được cho việc giải quyết vấn đề, giả sử, nếu xác định được bài toán thuộc dạng NP. Khi đó, thay vì tập trung nguồn lực giải quyết trường hợp chung cho bài toán, ta sẽ tập trung giải quyết từng trường hợp đơn lẻ, cụ thể. Đó là việc vô cùng tích cực bởi trong thực tế, với lượng thông tin và các điều kiện thu thập được, thì bài toán dễ hơn rất nhiều, và cùng với việc năng lực tính toán của máy tính ngày càng cao thì ta cũng có thể giải được các bài toán khó với một quy mô khá lớn phù hợp với thực tế.

2 Định nghĩa

2.1 Turing machine

Do các lớp bài toán được định nghĩa dựa trên nền tảng của Turing machine nên ta sẽ đề cập về khái niệm này đầu tiên.

Có khá nhiều loại Turing machine được đề cập tới bao gồm:

- Deterministic Turing machine
- Non-deterministic Turing machine
- Probabilistic Turing machine
- Quantum Turing machine
- ...

Tuy vậy, trong giới hạn bài viết ta chỉ đề cập cụ thể tới 2 loại máy đầu tiên, do chúng có liên quan mật thiết đến các lớp bài toán phía sau.

2.1.1 Deterministic Turing machine

Một Turing machine được biểu diễn bởi 1 nhóm $M = \langle \Sigma, \Gamma, b, Q, \delta, q_0, F \rangle$

Trong đó:

Q	tập hữu hạn, không rỗng gồm các trạng thái.
Γ	tập hữu hạn, không rỗng các kí tự trên đĩa(tape).
$b \in \Gamma$	kí tự trống (blank symbol), là kí tự duy nhất được xuất hiện vô số lần trên tape.
$\Sigma \subseteq \Gamma \setminus \{b\}$	tập hợp các kí tự input.
$\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$	là một hàm cục bộ (partial function) được gọi là một hàm chuyển (transition), L là dịch trái, R là dịch phải.
$q_0 \in Q$	là trạng thái ban đầu.
$F \subseteq Q$	là tập trạng thái kết thúc hoặc chấp nhận (accepting states). Nội dung 1 tape được xem là chấp nhận bởi turing machine M khi và chỉ khi nó kết thúc với một trạng thái thuộc F.

Một cấu hình (configuration) của M được biểu diễn bởi 1 chuỗi xqy , trong đó $x, y \in \Gamma^*$, và được hiểu là M đang ở trạng thái q với xy trên đĩa (tape), và đầu đọc đang đọc kí tự đầu tiên của y .

2.1.2 Non-deterministic Turing machine

2.2 Lớp bài toán P

Lớp P bao gồm tất cả các bài toán mà tồn tại một thuật toán có thể đưa ra lời giải của nó trong thời gian đa thức.

Trong đó P có nghĩa: Polynomial time.

Ví dụ: tìm số lớn nhất trong dãy số nguyên, cộng 2 số nguyên lớn.

2.3 Lớp bài toán NP

Lớp NP bao gồm tất cả các bài toán quyết định mà lời giải của nó có thể kiểm tra được trong thời gian đa thức.

Trong đó NP có nghĩa: Non-deterministic Polynomial time.

Ví dụ:

Bài toán tổng con cho một dãy số, hay tìm một dãy con có tổng bằng 0. Ta có thể kiểm tra kết quả bài toán bằng một vòng lặp nhưng để tìm ra được kết quả thì vô cùng khó khăn ngoại trừ cách duyệt hết tất cả, 2^N , tập con.

Bài toán Boolean Satisfiability cho một tập các biểu thức Boolean định nghĩa trên tập biến cho trước, liệu có tồn tại cách gán giá trị TRUE hay FALSE cho từng biến để tất cả các biểu thức đều mang giá trị TRUE hay không? Đây là một bài toán vô cùng quan trọng bởi nó được xem là bài toán thuộc lớp các bài toán khó nhất trong tất cả các bài toán NP. Điều này có nghĩa là nếu giải được bài toán này ta có thể giải được tất cả các bài trong lớp NP sử dụng thuật toán của nó. Hơn thế nữa, tất cả các bài toán trong lớp NP đều có thể rút gọn về bài toán này trong thời gian đa thức. Điều này mang ý nghĩa vô cùng lớn bởi nếu có thể tìm ra thuật toán trong thời gian đa thức cho bài toán này thì ta cũng có thể giải được tất cả các bài toán NP trong thời gian đa thức.

2.4 Lớp bài toán NP-Hard

Lớp NP-Hard bao gồm tất cả các bài toán mà "có độ khó tối thiểu bằng các bài toán khó nhất trong lớp NP", nhưng không nhất thiết nằm trong lớp NP.

Nói cách khác thì một bài toán A thuộc lớp NP-hard khi và chỉ khi với mỗi bài toán thuộc lớp NP, luôn tồn tại phép rút gọn trong thời gian đa thức về bài toán A.

Phát biểu hình thức:

NPH	lớp các bài toán NP-hard.
NP	lớp các bài toán NP.
\leq^P :	phép rút gọn trong thời gian đa thức. (ta sẽ đề cập tới định nghĩa của phép tính này sau)
Định nghĩa	$A \in NPH \Leftrightarrow \forall B \in NP, B \leq^P A$

Ví dụ:

Bài toán Knapsack Cho một túi sách với trọng lượng tối đa W . N đồ vật với trọng lượng w_i và giá trị v_i . Hãy tìm tập con các đồ vật để bỏ vào túi sao cho tổng giá trị là lớn nhất có thể và tổng trọng lượng không vượt quá W .

2.5 Lớp bài toán NP-Complete

Có nhiều cách để định nghĩa NP-Complete.

Một trong số đó dựa trên định nghĩa về NP-Hard và NP.

NPC lớp các bài toán NP-complete.

Định nghĩa $A \in NPC \Leftrightarrow A \in NPH \cap NP$

hoặc $A \in NPC \Leftrightarrow A \in NP \& \forall B \in NP, B \leq^P A$

Ví dụ:

Bài toán Knapsack - phiên bản bài toán quyết định Cho một túi sách với trọng lượng tối đa W . N đồ vật với trọng lượng w_i và giá trị v_i . Hỏi có thể bỏ các đồ vật vào túi sao cho tổng giá trị tối thiểu là V và trọng lượng không vượt quá W .

Bài toán Boolean Satisfiability bài toán này phát biểu tương tự như ở ví dụ phần 2.3.

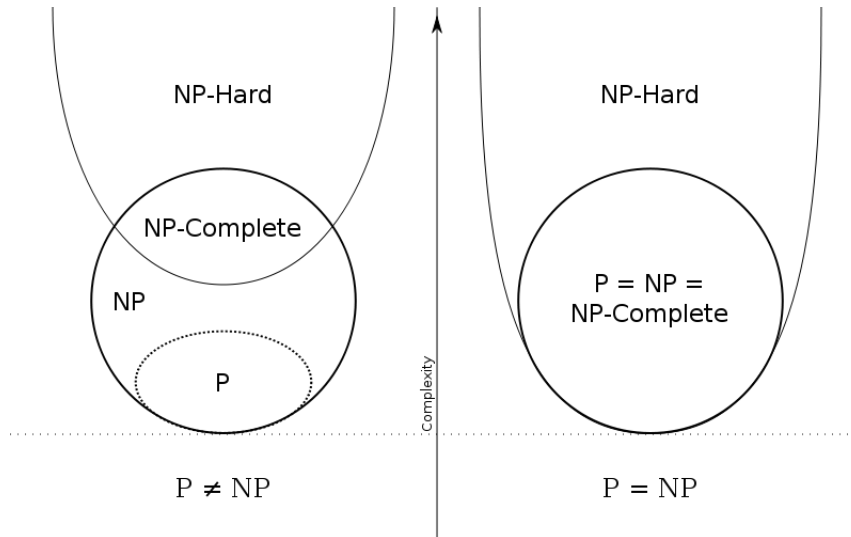
Bài toán Clique Cover Xét một đồ thị tổng quát, hỏi có cách nào để phân các đỉnh của đồ thị trên thành k đồ thị con đầy đủ hay không.

Ví dụ về phép rút gọn một bài toán về bài toán thuộc lớp NP-Complete:

Xét bài toán kiểm tra sắc số đồ thị (kiểm tra xem đồ thị có thể tô được bằng nhiều nhất k màu sao cho 2 đỉnh kề nhau tô màu khác nhau). Ta có thể biến đổi bài toán này về bài toán Clique Cover thông qua phép biến đổi trong thời gian đa thức như sau.

1. Xây dựng đồ thị bù của đồ thị cần kiểm tra sắc số. (Đồ thị G' gọi là bù của G nếu như $V_{G'} = V_G$ và 2 đỉnh u, v kề nhau trong G' khi và chỉ khi chúng không kề nhau trong G)
2. Giải bài toán Clique Cover trên G' .
3. Nếu câu trả lời là Yes, tức có thể phân G' thành k cliques thì ta cũng có thể tô đồ thị G bằng k màu, với các đỉnh trong cùng 1 clique của G' được tô cùng một màu. Do mỗi cặp đỉnh trong một clique của G' đôi một kề nhau, nên cũng có nghĩa là chúng đôi một không kề nhau trong G nên ta có thể tô cùng 1 màu.

3 Mỗi quan hệ giữa các lớp bài toán



Dựa vào biểu trên, ta có nhận xét: nếu tồn tại một bài toán thuộc lớp NPC mà cũng thuộc lớp P thì khi đó $NP = P$. Điều này cũng dễ dàng chứng minh được dựa vào định nghĩa của lớp NPC. Đây cũng là một cách tiếp cận với bài toán **P vs NP**.

4 Tạm kết

Dù có giải được bài toán P vs NP hay không thì những ý nghĩa mà nó mang lại cho thế giới là vô cùng lớn. Bởi trong quá trình tìm hiểu, đề xuất, và xây dựng phương pháp để giải quyết bài toán này thì thế giới đã làm giàu thêm kho tàng kiến thức của mình đồng thời giúp cho chúng ta có cái nhìn sâu sắc hơn trong lĩnh vực phân tích độ phức tạp thuật toán.