NP-completeness

Tạ Duy Hoàng

Đại học Bách Khoa Hà Nội

hoangduybk56@gmail.com

Ngày 6 tháng 11 năm 2017

Tổng Quan

1. P và NP
2. NP - Complete
3. Một số các thuật toán dùng để giải các bài toán thuộc lớp NP-hard

P và NP

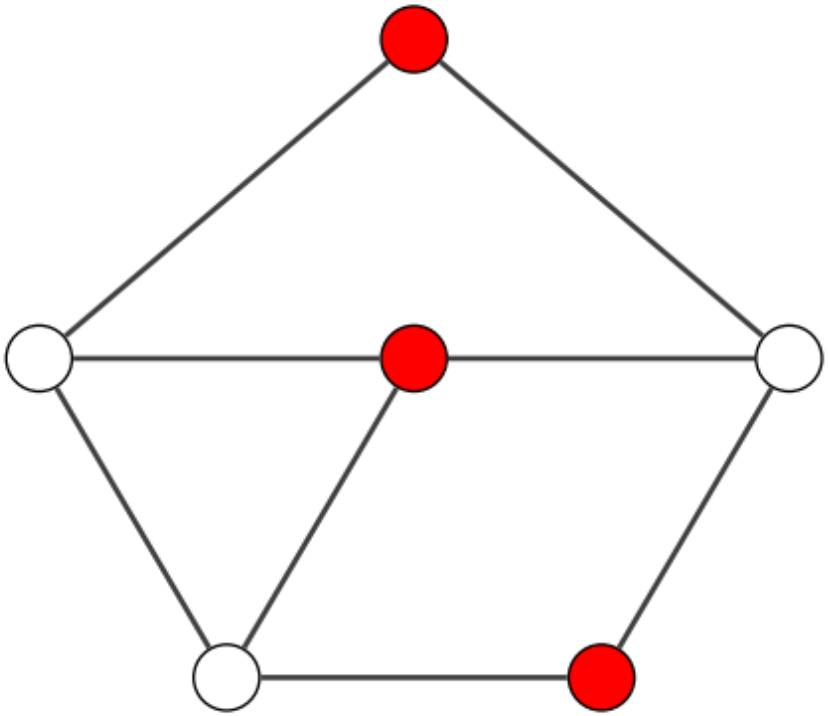
- Chúng ta thường mô tả độ phức tạp của một bài toán dựa vào các tài nguyên (resource) cần thiết để giải bài toán đó. Hai loại tài nguyên cơ bản là thời gian (time) và bộ nhớ (space), cụ thể chúng ta thường ký hiệu là $O(n), O(nlogn), O(n^2), \ldots$ hay tổng quát là $O(f(n))$.
  - Biến n ở đây chính là chiều dài của đầu vào (input) của bài toán (problem). Việc sử dụng n để phân tích độ phức tạp của bài toán dựa trên ý tưởng: bài toán càng lớn (dài) thì càng khó giải quyết. Chiều dài của đầu vào được định nghĩa là số bít dùng để biểu diễn đầu vào, và số bít này thì lại phụ thuộc vào cách mã hóa (encode) đầu vào.
  - Hàm $f(n)$ ở đây biểu diễn số **thao tác cơ bản** mà một thuật toán sử dụng để giải quyết bài toán có input chiều dài n. Tập thao tác cơ bản lại phụ thuộc vào mô hình tính toán (models of computation) mà ta sử dụng.
- Trong phân tích thuật toán, chúng ta thường sử dụng tiệm cận hơn là đếm chính xác số lượng các **thao tác cơ bản**.

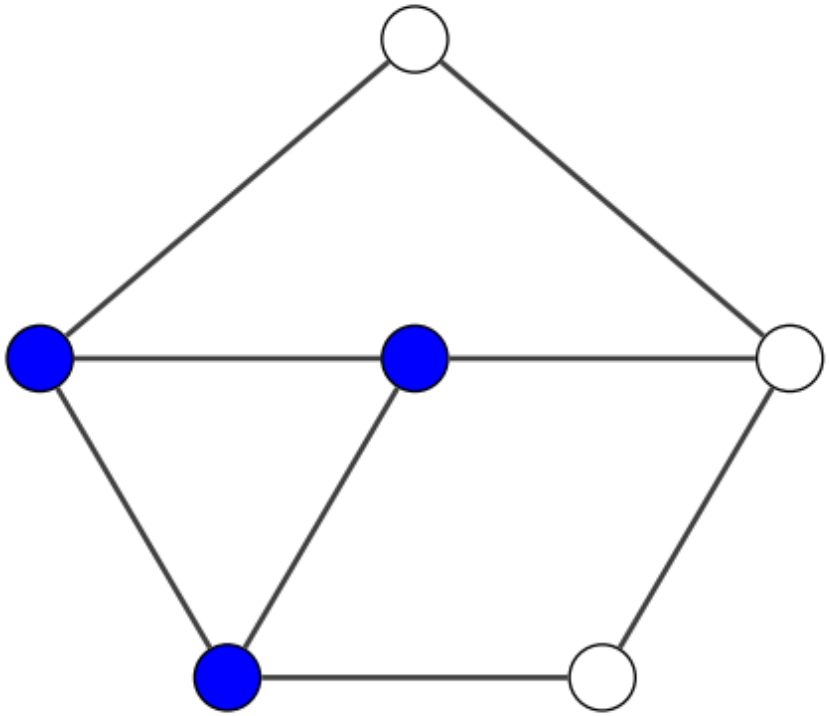
- Một bài toán bao giờ cũng có ít nhất hai phần là đầu vào (input) và đầu ra (output). Độ dài của input được coi như là biến để đo độ phức tạp của thuật toán giải bài toán.
- **Bài toán quyết định**: là bài toán mà đầu ra của nó chỉ có 1 bít (yes hoặc no).

Cho một đồ thị vô hướng $G = (V, E)$, một tập $S \subseteq V$ là:

- independent set: Không có hai đỉnh nào trong S được nối bởi một cạnh.
- clique: mỗi cặp đỉnh trong S được nối bởi một cạnh trong G.

Ví dụ về independent set và clique





Bài toán Independent Set và Clique

Independent Set

Instance: Một đồ thị G và số nguyên k.
Question: Đồ thị G có chứa một independent set có lực lượng $\geq k$?

Clique

Instance: Một đồ thị G và số nguyên k.
Question: Đồ thị G có chứa một clique có lực lượng $\geq k$?

Định Nghĩa 1

P problem: Một bài toán quyết định được gọi là thuộc lớp **P** nếu tồn tại một thuật toán giải bài toán trong thời gian $O(n^c)$ với một hằng số c không phụ thuộc vào kích thước đầu vào n

Ta sẽ dùng $poly(n)$ để thay thế cho $O(n^c)$.

- Một đặc điểm trung của rất nhiều các bài toán quyết định khó giải là nếu ai đó cho chúng ta lời giải của bài toán đó, việc kiểm tra xem đó có phải là lời giải đúng hay không sẽ **dễ hơn** việc trực tiếp giải bài toán đó.
- Bài toán **independent set:** Nếu ai đó đưa cho ta một tập đỉnh X của đồ thị và nói rằng tập đỉnh đó là độc lập của đồ thị có kích thước ít nhất là k, ta có thể kiểm tra trong $poly(n)$ xem X có phải tập độc lập hay không và $|X| \geq k$ hay không.

- Tập X trên gọi là một **bằng chứng**(certificate) cho thấy input của bài toán có output Yes. Hai tính chất quan trọng của X mà ta có thể rút ra
  - 1. **Ngắn gọn**: Ta mô tả X sử dụng tối đa $poly(n)$ bít vì $|X| \leq n$.
  - 2. **Dễ kiểm tra**: Ta có thể kiểm tra xem X có thỏa mãn điều kiện là một tập độc lập có kích thước k hay không trong thời gian $poly(n)$.
- Ta gọi X là một **bằng chứng ngắn gọn dễ kiểm tra** của bài toán independent set.

Định Nghĩa 2

NP problem : Một bài toán quyết định được gọi là thuộc lớp **NP** nếu tồn tại một **bằng chứng ngắn gọn dễ kiểm tra** cho câu trả lời Yes của bài toán đó.

- Theo định nghĩa trên, để chứng minh một bài toán thuộc lớp **NP** ta phải:
  - 1. Chỉ ra bằng chứng ngắn gọn cho đầu vào Yes.
  - 2. Chỉ ra một thuật toán để kiểm tra bằng chứng đó trong thời gian **poly(n)**

- Về mặt trực quan, một bài toán dễ giải thì cũng dễ kiểm tra lời giải, do đó $P \subset NP$, câu hỏi trung tâm trong lý thuyết tính toán.
- Chứng minh $P \subseteq NP$ theo định nghĩa bằng chứng ngắn ngọn dễ kiểm tra

P vs NP: Liệu **P** và **NP** có phải là hai lớp hoàn toán khác nhau hay không?

Một vài ví dụ về các bài toán thuộc lớp NP

SAT

SAT(Satisfiability). Cho một biểu thức logic $\varphi(n, m)$ gồm n biến $x_1, \ldots, x_n$ có dạng hội (conjuction) của m mệnh đề(clause) $C_1, \ldots, C_m$, trong đó mỗi mệnh đề C_i là tuyển (disjunction) của các toán hạng (literal). Mỗi literal là một biến x_j hoặc phủ định của nó $\overline{x_j}$. Một phép gán(assigment) của $\varphi(n, m)$ là một cách gán cho mỗi biến x_i một giá trị true hoặc false. Giá trị của một phép gán là giá trị của biểu thức (True hoặc False) khi ta thay giá trị tương ứng của biến vào biểu thức. Xác định xem có tồn tại hay không một phép gán của biểu thức $\varphi(n, m)$ có giá trị true.

Ví Dụ:

$$\varphi(3,3) = (x \vee y \vee z) \wedge (\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z)$$
$$\varphi(2,4) = (\overline{x} \vee y) \wedge (x \vee \overline{y}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{y})$$

Trong ví dụ trên, $\varphi(3,3)$ có phép gán $(x, y, z) = (true, true, true)$ có giá trị true, $\varphi(2,4)$ không có phép gán nào cho giá trị true.

3SAT

Cho một biểu thức logic $\varphi(n, m)$ trong đó mỗi mệnh đề chỉ gồm không quá 3 literals, xác định xem có tồn tại hay không một phép gán của biểu thức $\varphi(n, m)$ có giá trị True.

SUBSETSUM

Cho một tập gồm n số nguyên $A = \{a_1, a_2, ..., a_n\}$ và một số nguyên dương T, Có tồn tại 1 tập $S \subseteq A$ sao cho :

$$\sum_{a \in S} a = T$$

Ví dụ $A = \{1, 3, 11, 6, 7, 8\}$ và $T = 10$, tập $S = \{1, 3, 6\}$ có tổng bằng T.

KNAPSACK

Cho n đồ vật, trong đó đồ vật thứ i có trọng lượng w_i và giá trị v_i và một cái ba lô có khả năng mang được trọng lượng tối đa W. Liệu ta có thể đựng được đồ vật trong ba lô (mà không quá giới hạn trọng lượng của ba lô) có tổng giá trị ít nhất V hay không?

HAMILTONIAN CYCLE

Cho một đồ thị có hướng $G = (V, E)$. Một chu trình C trong đồ thị được gọi là chu trình **Hamilton** nếu chu trình này đi qua mỗi đỉnh đúng một lần. Xác định xem có tồn tại một chu trình Hamilton trong $G = (V, E)$ hay không?

TSP

Cho một đồ thị vô hướng $G = (V, E)$, trong đó mỗi cạnh $e \in E$ có một trọng số không âm $w(e)$, và một số nguyên K. Một đường đi đóng là dãy đỉnh $v_1, v_2, \ldots, v_{l-1}, v_l = v_1$. Có tồn tại hay không một đường đi đóng trong G đi qua tất cả các đỉnh đúng một lần (trừ đỉnh đầu tiên) mà có tổng trọng số các cạnh trên đường đi không quá K

NP - Complete

- Trong mục này, chúng ta sẽ xem một cách tiếp cận đối với câu hỏi **P vs NP** thông qua một lớp các bài toán khó nhất trong số các bài toán trong lớp **NP**, những bài toán đó được gọi là bài toán thuộc lớp **NP-complete**.
- Làm thế nào để biết được bài toán nào là khó nhất trong số vô hạn các bài toán trong **NP**?
- Cook và Levin đã độc lập đưa ra câu trả lời cho câu hỏi này, câu trả lời trở thành **định lý Cook-Levin**. Định lý này (cùng một số công trình khác) mang về cho Cook giải Turing
- Công cụ sử dụng trong chứng minh của định lý Cook-Levin chính là **quy dẫn** (reduction)

Quy dẫn (Reduction)

- Một input x của một bài toán quyết định A được gọi là một **Yes-instance** nếu đầu ra tương ứng của input này là **Yes**. Ngược lại, ta gọi x là một **No-instance** của bài toán A. Ta sẽ dùng $|x|$ để kí hiệu chiều dài của input x.

Karp Reduction

Một bài toán quyết định A gọi là quy được về (reducible to) bài toán quyết định B trong thời gian đa thức nếu với mỗi input x của bài toán A, tồn tại một giải thuật M, trong thời gian $poly(|x|)$, tạo ra $M(x)$ sao cho x là một Yes-instance của A khi và chỉ khi $M(x)$ là một Yes-instance của bài toán B. Ta sẽ kí hiệu:

$$A \preceq_{Karp} B$$

Tính chất bắc cầu của quy dẫn Karp: Nếu $A \preceq_{Karp} B$ và $B \preceq_{Karp} C$ thì $A \preceq_{Karp} C$

Lemma 1

Nếu $A \preceq_{Karp} B$ và $B \in \mathbf{P}$ thì $A \in \mathbf{P}$

Chứng minh: Gọi x là một đầu vào của bài toán A. Để xác định xem x có phải là một Yes-instance của A hay không, ta sẽ:

1. Áp dụng thuật toán quy dẫn M để biến đầu vào x thành đầu vào $M(x)$ của bài toán B. Do thuật toán quy dẫn có thời gian đa thức, chiều dài của $|M(x)|$ cũng là đa thức, i.e, $|M(x)| = poly(|x|)$.

2. Xác định xem $M(x)$ có phải là một Yes-instance của B hay không. Việc xác định này có thể thực hiện được trong thời gian $poly(|M(x)|)$ vì $B \in \mathbf{P}$. Ta có $poly(|M(x)|) = poly(poly(|x|)) = poly(|x|)$, cả hai bước trên đều có thể thực hiện được trong thời gian đa thức (đpcm).

Ý nghĩa của quy dẫn: Quy dẫn cung cấp cho chúng ta một công cụ rất đẹp để định lượng "độ khó" giữa hai bài toán. Nếu $A \preceq_{Karp} B$ thì **Lemma 1** cho chúng ta biết bài toán A sẽ dễ hơn bài toán B, theo nghĩa, nếu ta giải được B trong thời gian đa thức thì ta cũng giải được A trong thời gian đa thức. Điều ngược lại **có thể sẽ không đúng**.

Ta đã có trong tay công cụ để xác định độ khó giữa các bài toán. Giờ ta có thể định nghĩa một lớp các bài toán "khó", khó hơn tất cả các bài toán trong lớp **NP**

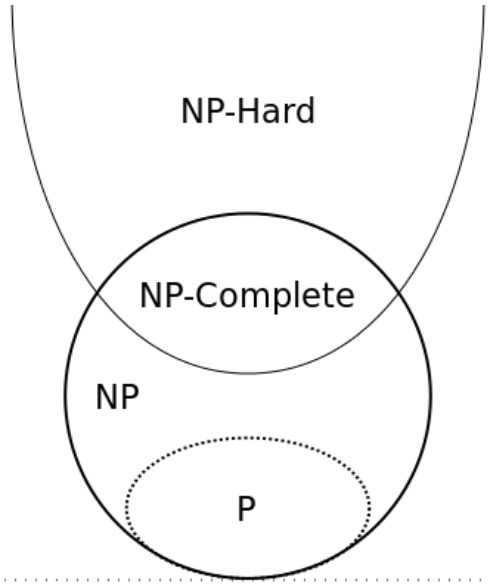
NP-hard: Một bài toán B được gọi là thuộc lớp các bài toán **NP-hard** nếu **với mọi** bài toán $A \in$ **NP**, ta có:

$$A \preceq_{Karp} B$$

Nếu bài toán B trong định nghĩa trên cũng thuộc lớp **NP**, ta nói B thuộc lớp **NP-complete**.

NP-complete = NP $\cap$ NP-hard

- Theo định nghĩa trên, **NP-complete** chỉ là một lớp bài toán con của lớp **NP-hard**. Lớp **NP-hard** còn bao gồm những bài toán mà kể cả cho máy tính vô hạn thời gian, nó cũng không giải được. Ngược lại, tất cả các bài toán thuộc lớp **NP** ta đều có thể giải được trong một thời gian hữu hạn; cụ thể là thời gian lũy thừa.
- Cũng theo định nghĩa trên, một bài toán thuộc **NP-complete** thì cũng thuộc **NP-hard**. Do đó, nó là một bài toán "khó nhất" trong số các bài toán trong lớp **NP**. Nếu ta có thể giải được một bài toán nào đó trong lớp **NP-complete** trong thời gian đa thức thì ta có thể giải được tất cả các bài toán trong lớp **NP** trong thời gian đa thức.



Lớp bài toán **NP-complete** theo định nghĩa ở trên thực sự là một lớp bài toán thú vị. Tuy nhiên, liệu một bài toán **NP-complete** có tồn tại hay không? Nếu tồn tại thì nó trông như thế nào? **Cook** và **Levin** chứng minh rằng không những tồn tại một bài toán **NP-complete** mà còn tồn tại một bài toán NP-complete rất "đẹp".

Cook-Levin Theorem

$SAT \in$ **NP - complete**

Sau định lý Cook-Levin, Richard Karp chứng minh 21 bài toán khác cũng thuộc lớp **NP-complete**. Một vài trong số đó là các bài toán NP mà ta đã định nghĩa ở trên . Các chứng minh của **Karp** đều dựa trên bổ đề sau:

Lemma 2

Nếu A là một bài toán **NP-complete** và B là một bài toán **NP** thỏa mãn $A \preceq_{Karp} B$ thì B là một bài toán **NP-complete**.

Theorem

$3SAT \in NP - complete$

Chứng minh: Dễ thấy bài toán $3SAT$ thuộc lớp NP, ta chỉ cần chứng minh

$$SAT \preceq_{Karp} 3SAT$$

Gọi $\varphi(n, m)$ là một input SAT. Ta sẽ biến đổi, trong thời gian poly(n+m), $\varphi(n, m)$ thành một biểu thức $\varphi'(n', m')$ sao cho $\varphi'(n', m') \in 3SAT$ và $\varphi(n, m)$ thỏa mãn được (satisfiable) khi và chỉ khi $\varphi'(n', m')$ thõa mãn được

Lấy một mệnh đề C trong $\varphi(n, m)$ mà mệnh đề này có k biến với $k \geq 4$, các biểu thức 3 biến chỉ cần giữ nguyên, giả sử

$C = (x_1 \vee \overline{x_2} \vee \overline{x_3} \vee x_4 \vee ... \vee x_k)$ thêm một biến z không xuất hiện trong $\varphi(n, m)$ và thay C bằng :

$$(x_1 \vee \overline{x_2} \vee z) \wedge (\overline{z} \vee \overline{x_3} \vee x_4 \vee ... \vee x_k)$$

Gọi $\varphi_1(n_1, m_1)$ với $n_1 = n + 1$ và $m_1 = m + 1$

Theorem

INDEPENDENTSET $\in$ **NP-complete**

Chứng minh: Ta sẽ chỉ ra $3SAT \preceq_{Karp} INDEPENDENTSET$, tức là cho φ là một công thức $3SAT$ nào đó (giả sử mỗi biểu thức có đúng 3 toán hạng) ta phải xây dựng đồ thị $G = (V, E)$ và số nguyên K sao cho : G có tập độc lập có lực lượng ít nhất là K khi và chỉ khi φ là nhận giá trị *True*.

- Gọi K là số mệnh đề trong φ.
- Mỗi mệnh đề trong φ sẽ tương ứng với một nhóm gồm các đỉnh được nối với nhau đôi một, mỗi đỉnh tương ứng với một toán hạng.
- Hai đỉnh ở 2 nhóm khác nhau được nối với nhau khi chúng tương ứng với 2 toán hạng đối nghịch nhau (dạng x và $\overline{x}$)

Ta cần chỉ ra rằng : φ nhận giá trị True khi và chỉ khi G có tập độc lập S với kích thước K

Quy dẫn bài toán 3SAT và bài toán Independent Set

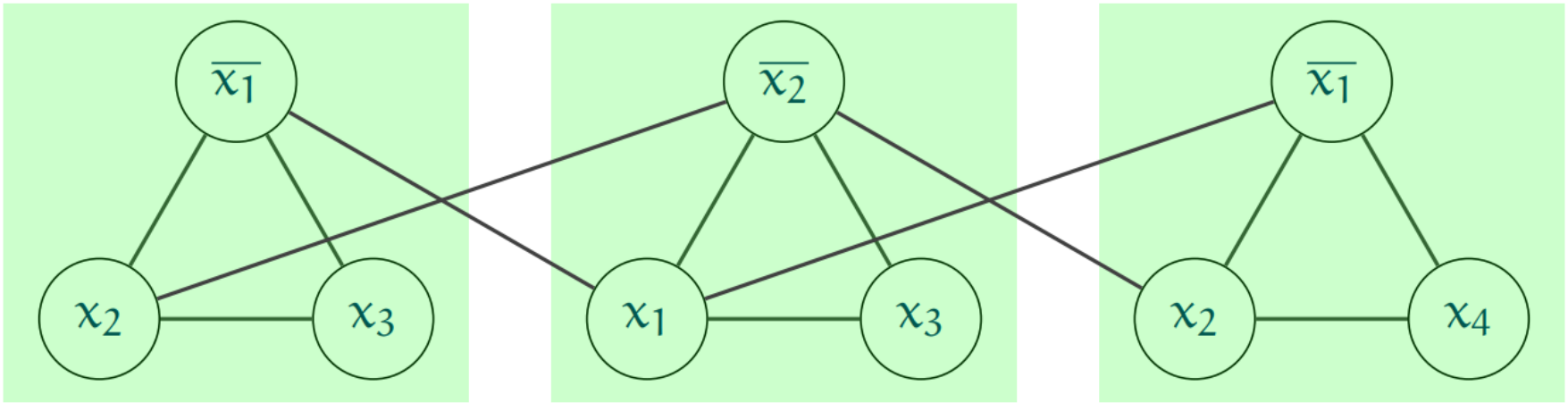


Figure: Graph for $\varphi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4)$

- $\Rightarrow$ Nếu có bộ dữ liệu **True** φ, thì rõ ràng trong mỗi mệnh đề phải có ít nhất một toán hạng nhận giá trị **True**. Chọn trong mỗi mệnh đề một toán hạng nhận giá trị **True**. Tập S gồm những đỉnh tương ứng với các toán hạng được chọn cho ta tập độc lập lực lượng K. Bởi vì ta sử dụng một đỉnh trong mỗi mệnh đề nên S chỉ có thể không là tập độc lập nếu nó bao gồm các toán hạng phủ định nhau, điều đó không thể xảy ra vì không thể gán 2 toán hạng phủ định nhau cùng nhận giá trị *True*.

- $\Leftarrow$ Giả sử G có tập độc lập S kích thước K. Từ cách xây dựng G suy ra các đỉnh trong S phải được chọn từ các nhóm đỉnh khác nhau, mỗi nhóm chọn đúng 1 đỉnh. Xét bộ giá trị φ trong đó tất cả các toán hạng tương ứng với các đỉnh trong S có giá trị **True** còn các toán hạng còn lại gán giá trị **False**. Rõ ràng φ là giá trị tương thích do không có 2 toán hạng phủ định nhau đồng thời có mặt trong cùng một mệnh đề, φ nhận giá trị **True**.

Cho một đồ thị vô hướng $G = (V, E)$, một vertex cover là một tập con S của V sao cho với mỗi cạnh e của G thì e có ít nhất một đầu mút trong S

Vertex Cover problem

Instance: Một đồ thị G và số nguyên k
Question: G có một vertex cover có lực lượng nhỏ hơn k?

Theorem

Bài toán Vertex cover thuộc lớp $NP - complete$

Chứng minh:

- **Vertex cover** $\in$ NP
- **Independent Set** $\preceq_{Karp}$ **Vertex cover**

Theorem

Nếu $G = (V, E)$ là một đơn đồ thị vô hướng, thì S là một independent set của G khi và chỉ khi $V - S$ là một vertex cover

Chứng minh.

- $\Rightarrow$

  Giả sử S là một independent set, và $e = (u, v)$ là một cạnh nào đó của $G \Rightarrow u$ và v không cùng nằm trong S. Vì vậy ít nhất u hoặc v hoặc cả 2 nằm trong $V - S$, do đó $V - S$ là một vertex cover.

- $\Leftarrow$

  Giả sử $V - S$ là một vertex cover, và cho $u, v \in S$. Không thể tồn tại cạnh giữa u và v (ngược lại thì cạnh đó sẽ không được cover bởi tập $V - S$), do đó S là independent set

Set cover

Instance: Cho một tập U, m tập $S_1, \ldots, S_m$ là tập con của U và số nguyên k

Question: Có tồn tại một họ F (với các tử của F là lấy trong $S_1, \ldots, S_m$) có lực lượng $\leq k$ và hợp của tất cả các tập hợp trong F bằng U?

Theorem

Set cover thuộc lớp $NP - complete$

Chứng minh: Quy dẫn từ bài toán Vertex cover.

Theorem

Bài toán **Hamiltonian Cycle** thuộc lớp *NP – complete*

Chứng minh.

- Quy dẫn từ bài toán Vertex cover
- Quy dẫn từ bài toán 3SAT

Theorem

Hamiltonian Cycle $\preceq_{Karp}$ **TSP**

Chứng minh.

Cho một **Hamiltonian Cycle** của $G = (V, E)$, tạo n thành phố với khoảng cách như sau:

$$d(u, v) = \begin{cases} 1 & \text{Nếu (u,v)} \in E \\ 2 & \text{Nếu (u,v)} \notin E \end{cases}$$

TSP có một tour có độ dài $\leq n$ khi và chỉ khi G có Hamiltonian Cycle

Weakly NP-complete problems

□ : Ta sẽ xét một số bài toán **NP-complete** khi mà đầu vào là các số nguyên lớn. Ta gọi các bài toán **NP-complete** kiểu này là các bài toán *Weakly NP-complete problems*

SUBSETSUM

Cho một mảng n phần tử $X[1, 2, ..., n]$, trong đó phần tử thức i có giá trị $X[i]$ và một số T. Có tồn tại hay không một tập con các phần tử của mảng X sao cho tổng giá trị của chúng bằng T.

Sử dụng quy hoạch động ta có thể giải bài toán SUBSETSUM với thời gian $O(nT)$ nhưng số nguyên T có thể biểu diễn bằng $O(logT)$ bít,do đó, nó có chiều dài $l = O(logT)$. Như vậy thời gian $O(nT) = O(n2^{logT}) = O(n2^l)$ chính là một hàm lũy thừa, trong mục này ta sẽ chứng minh khi T rất lớn , bài toán SUBSETSUM là một bài toán **NP-complete**

PARTITION

Cho một mảng n phần tử $Y[1, 2, \ldots, n]$, trong đó phần tử thứ i có giá trị $Y[i]$. Có tồn tại hay không một cách chia tập X làm đôi sao cho tổng giá trị của hai tập con bằng nhau.

KNAPSACK

Cho n đồ vật, trong đó đồ vật thứ i có trọng lượng w_i và giá trị v_i và một cái ba lô có khả năng mang được trọng lượng tối đa W. Liệu ta có thể đựng được đồ vật trong ba lô (mà không quá giới hạn trọng lượng của ba lô) có tổng giá trị ít nhất V hay không?

VERTEXCOVER

Cho một đồ thị vô hướng $G = (V, E)$ và một số nguyên K. Có tồn tại hay không một tập phủ đỉnh có kích thước không quá K trong đồ thị? Một tập đỉnh $A \subseteq V$được gọi là một tập phủ đỉnh (vertex cover) nếu như với mọi cạnh e của đồ thị, ít nhất một trong hai đầu mút của nó thuộc A.

Theorem

SUBSETSUM $\in$ **NP-complete**

Chứng minh: Ta sẽ quy dẫn từ bài toán **VERTEXCOVER** như sau. Gọi n, m lần lượt là số đỉnh và cạnh của đồ thị. Ta thực hiện hai bước sau :
1. Đánh số các cạnh của đồ thị từ 0 tới $m-1$. Cạnh thứ i ta gán cho nó trọng số $X[e_i] = 10^i$ và gán cho 2 đỉnh đầu mút của nó, mỗi đỉnh một trọng số 10^i
2 . Với mỗi đỉnh v của đồ thị, ngoài trọng số nó nhận được từ các cạnh kề với nó từ bước 1, ta cộng thêm cho nó một trọng số 10^m. Gọi $X[v]$ là tổng trọng số cuối cùng của đỉnh v.
Gọi E_v là tập các cạnh kề với đỉnh v. Ta có nhận xét về tổng trọng số mà đỉnh v nhận được sau 2 bước trên là:

$$X[v] = 10^m + \sum_{j \in E_v} 10^j \qquad (1)$$

Nếu viết $X[v]$ trong cơ số 10, ta có thể hiểu đại lượng $X[v]$ như sau: chữ số thứ $m+1$ của $X[v]$ luôn là số 1 và chữ số thứ j của $X[v]$ là 1 khi và chỉ khi v kề với cạnh $j+1$

Mảng X của chúng ta bao gồm tất cả các đỉnh và cạnh, có kích thước $m+n$. Cạnh i, $0 \leq i \geq m-1$ có giá trị 10^i, và đỉnh v, $1 \leq v \geq n$ có giá trị như trong phương trình (1). Ta thiết lập :

$$T = K.10^m + 2 * \sum_{i=0}^{m} 10^i \qquad (2)$$

Ta sẽ chứng minh $G = (V, E)$ có một tập phủ đỉnh kích thước K khi và chỉ khi X có một tập con có tổng giá trị T định nghĩa trong (2).

$\square \Rightarrow$: Gọi A là tập phủ đỉnh của đồ thị với kích thước K. Ta xây dựng tập con S của X bao gồm: (i) tất cả các đỉnh trong A và (2) các cạnh của đồ thị có **đúng** một đầu mút trong A. Do A là một tập phủ đỉnh, mỗi cạnh có **ít nhất** một đầu mút trong A nên tổng các phần tử trong S có giá trị bằng T

$\square \Leftarrow$: Gọi S là tập con của X có tổng giá trị bằng T. Do tổng trọng số các cạnh của đồ thị là $\sum_{i=0}^{m-1} 10^i$, nhỏ hơn 10^m, tập S gồm đúng K đỉnh (và có thể có thêm một số cạnh). Gọi A là tập các đỉnh trong S. Ta chỉ còn phải chứng minh mọi cạnh đều có **ít nhất** một đầu mút trong A. Giả sử có 1 cạnh có số thứ tự l với $0 \leq l \geq m-1$ không có đầu mút trong A, cạnh này có trọng số 10^l. Do đó, cạnh này đóng góp 10^l trong T, tuy nhiên do cạnh này không kề với đầu mút nào trong A và $2 * \sum_{i=0}^{l-1} 10^i$ nhỏ hơn 10^l, T chỉ chứa tối đa 10^l, trái với định nghĩa trong (2) là T chứa $2 * 10^l$.

- **Quy dẫn bài toán** $3SAT$ **to SUBSET SUM**.

Theorem

PARTITION thuộc lớp NP-complete

Chứng minh 1

Bài toán PARTITION thực ra là trường hợp đặc biệt của bài toán SUBSETSUM khi $X[i] = Y[i]$ và $T = \frac{\sum_{i=1}^{n} Y[i]}{2}$

Chứng minh 2.

Ta đặt $C = \sum_{i=1}^{n} X[i]$. Gọi Y là tập gồm $n+2$ phần tử, trong đó $Y[i] = X[i]$ với $1 \leq i \leq n$ và $Y[n+1] = C^2 + T$, $Y[n+2] = C^2 + C - T$. Ta sẽ chứng minh tập X có một tập con có tổng giá trị T khi và chỉ khi Y có thể chia thành hai tập con có tổng giá trị bằng nhau ($Y[n+1]$ và $Y[n+2]$ không thể cùng thuộc một tập con)

Hai chứng minh trên chứng minh nào **đúng** cho định lý?

- Packing/covering problems: **Set Cover**, **Vertex Cover**, **Independent Set**
- Constraint satisfaction problems: **SAT**, **3SAT**
- Sequencing problems: **Hamiltonian Cycle**, **TSP**
- Partitioning problems: **3D-Matching**, **3-Color**
- Numerical problems: **Subset Sum**, **Partition**

Nhiều bài toán NP đã biết thuộc P hoặc $NP - complete$
Notable exceptions: **Factoring, Graph Isomorphism, Nash Equilibrium**

Theorem(Ladner 1975), NP-intermediate

Nếu $P \neq NP$ thì tồn tại một bài toán trong NP mà không thuộc cả lớp P và NP

Karp's 21 NP-complete problems

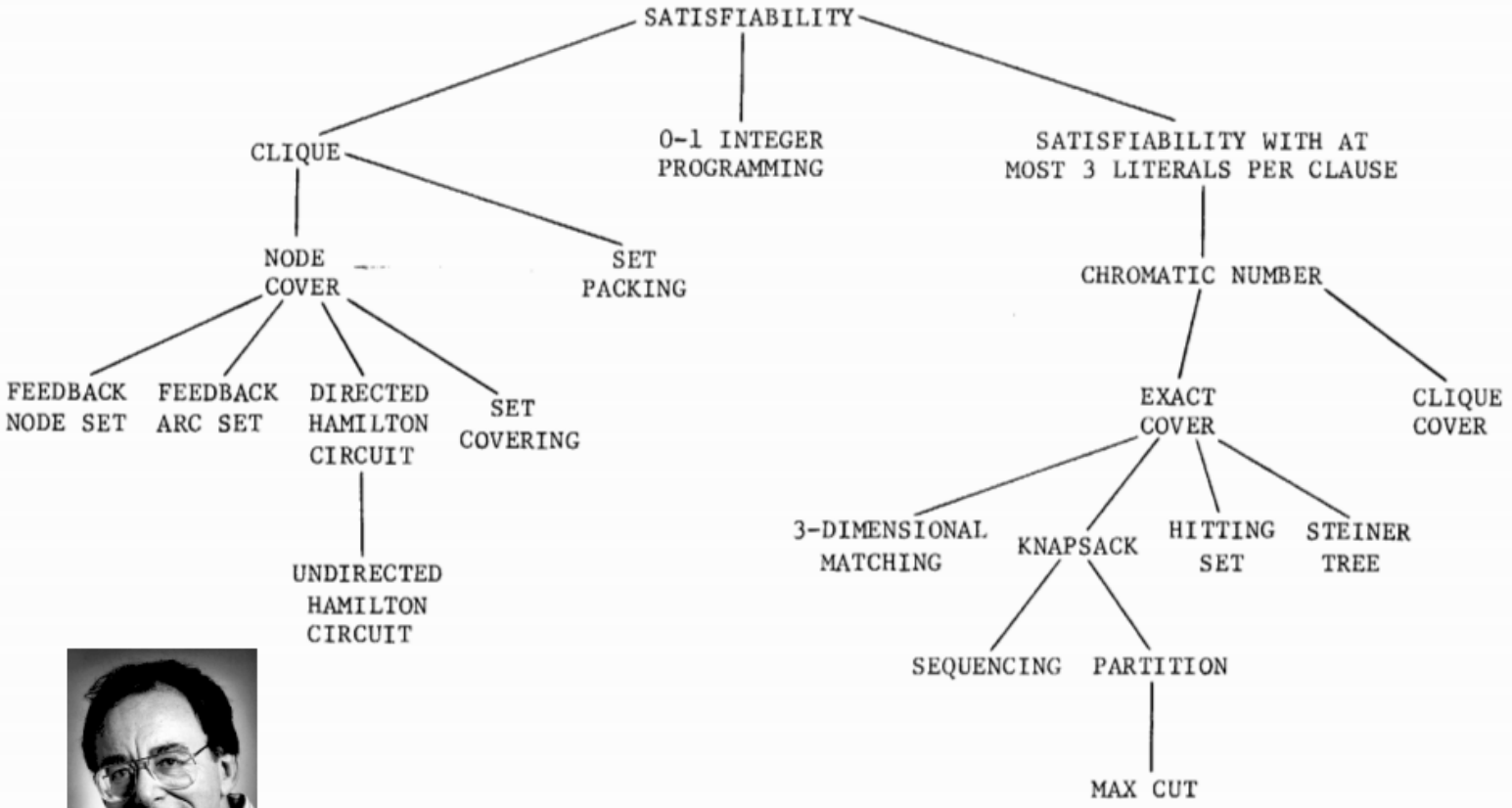


FIGURE 1 - Complete Problems

Dick Karp (1972)
1985 Turing Award

Small Difference ? Big Differences

P	NP-**complete**
shortest path	longest path
min cut	max cut
2-satisfiability	3-satisfiability
planar 4-colorability	planar 3-colorability
bipartite vertex cover	vertex cover
matching	3d-matching
linear programming	integer linear programming

Các thuật toán

- Thuật toán xấp xỉ
- Fixed Parameter Algorithms
- Thuật toán xác suất