LỊCH SỬ TÓM TẮT VỀ NP-ĐẦY ĐỦ, 1954–2012

David S. Johnson
(Người dịch: Hoàng Anh Đức)

Ngày 13 tháng 2 năm 2020

GIỚI THIỆU CỦA NGƯỜI DỊCH

Đây là bản dịch tiếng việt của bài báo [David S. Johnson. A Brief History of NP-Completeness, 1954–2012, *Documenta Math.* Extra Volume: Optimization Stories (2012), 359–376][a]. Bản dịch được hoàn thành bởi Hoàng Anh Đức và sử dụng mẫu LaTeX của Tạp chí Epsilon[b]. Các bức ảnh trong bản dịch được chụp lại từ bản gốc tiếng Anh. **Mọi sai sót trong bản dịch này hoàn toàn là do hạn chế về kiến thức của người dịch.** Mọi góp ý xin gửi về anhduc.hoang1990@gmail.com.

Từ khoá: NP-đầy đủ (NP-completeness), thời gian đa thức (polynomial time), các thuật toán xấp xỉ (approximation algorithms), bài toán đóng thùng (bin packing), giả thuyết các trò chơi độc đáo (unique games conjecture)

[a] https://www.math.uni-bielefeld.de/documenta/vol-ismp/50_johnson-david.pdf
[b] https://www.facebook.com/TapchiEpsilon/

Năm 2012 đánh dấu kỷ niệm 40 năm bài báo "Reducibility among combinatorial problems" của Richard Karp [37] được công bố. Đây là bài báo đầu tiên minh họa tầm ứng dụng rộng khắp của một khái niệm ngày nay được biết đến với cái tên NP-đầy đủ. Khái niệm này đã được đề xuất một năm trước đó một cách độc lập bởi Stephen Cook và Leonid Levin. 2012 cũng là năm kỷ niệm sinh nhật lần thứ 100 của Alan Turing, người phát minh mô hình đặt nền móng cho khái niệm NP-đầy đủ mà ngày nay được biết đến như là "máy Turing". Trong bài viết này, tôi sẽ phác họa một cách khái quát về lịch sử và tiền sử của NP-đầy đủ (với ảnh minh họa), và đưa ra một bản tóm tắt ngắn gọn của cá nhân tôi về những tiến bộ về mặt lý thuyết trong 40 năm qua và tầm ảnh hưởng (hay những thiếu sót) của chúng trong thực hành cũng như trong lý thuyết tối ưu. Tôi giả thiết rằng độc giả quen thuộc với các khái niệm cơ bản NP-đầy đủ, P, và NP, mặc dù tôi hy vọng rằng câu chuyện tôi kể vẫn sẽ thú vị ngay cả với những người chỉ nhớ một cách mơ hồ về các định nghĩa này.

Tiền sử mới

Khi cuốn sách Garey & Johnson *Computers and Intractability: A Guide to the Theory of NP-Completeness* [23] được viết vào cuối những năm 1970, những tài liệu về lý thuyết chỉ được tham

khảo từ năm 1965. Cụ thể, chúng tôi trích dẫn các bài báo của Cobham [13] và Edmonds [18], những người đầu tiên xác định rằng lớp các bài toán giải được trong thời gian đa thức có liên quan đến khái niệm tính giải được hiệu quả (efficient solvability) và đáng giá nghiên cứu. Chúng tôi cũng trích dẫn bài báo thứ hai của Edmonds [17], trong đó, theo một ý nghĩa nhất định, giới thiệu thứ mà về sau gọi là lớp NP, bằng cách đề xuất khái niệm về một bài toán có một "đặc tính tốt."

Tuy nhiên, hóa ra là, có hai nhà toán học lỗi lạc đã đụng chạm đến các vấn đề liên quan đến NP-đầy đủ từ hơn một thập kỷ trước đó, qua các bức thư tay cá nhân mà rất nhiều năm sau mới được công bố. Bức thư đầu tiên được biết đến (và là bức thư thứ hai được viết) là lá thư của Kurt Gödel gửi cho John von Neumann, cả hai người sau đó đều làm việc ở Viện nghiên cứu cao cấp (Institute for Advanced Study) ở Princeton, New Jersey. Gödel có lẽ nổi tiếng nhất với "Các định lý bất toàn" ("Incompleteness Theorems") ông đề xuất năm 1931 về logic toán học. Lá thư của ông, viết bằng tiếng Đức ngày 20 tháng 03 năm 1956, không được công khai cho đến năm 1989, khi Juris Hartmanis công bố một bản dịch kèm theo nhận xét [27].

Trong lá thư này, Gödel đầu tiên xét một bài toán tìm các chứng minh (proof) trong một hệ chứng minh (proof system) cho trước: Cho một công thức logic cấp một (first order logic) F và một số nguyên n, có tồn tại hay không một chứng minh của F có độ dài không vượt quá n? Gọi A là một máy Turing giải bài toán trên, và, theo Gödel, gọi $\psi_A(F, n)$ là số các bước mà A cần thực hiện khi giải một trường hợp (instance) với công thức F và chặn trên n. Bây giờ, gọi $\phi_A(n)$ là giá trị trong trường hợp xấu nhất của $\psi_A(F, n)$ trên tất cả các công thức F có độ dài n. Chú ý rằng một máy Turing A thực hiện tìm kiếm vét cạn (exhaustive search) sẽ cho ra giá trị của $\phi_A(n)$ không tệ hơn hàm mũ theo n. Gödel chỉ ra rằng sẽ thật tuyệt vời nếu có một máy A với $\phi_A(n) = O(n)$ hay thậm chí $O(n^2)$, với chú ý rằng sự tăng tốc này cũng đã được nhận xét trước đó cho bài toán tính ký hiệu thặng dư bậc hai (quadratic residue symbol). Cuối cùng, ông hỏi liệu có thể cải tiến "một cách mạnh mẽ đến mức nào" so với phương pháp tìm kiếm vét cạn khi giải các bài toán tổ hợp, đặc biệt là trong bài toán kiểm tra tính nguyên tố (một bài toán mà việc xác định độ phức tạp trong trường hợp xấu nhất là một bài toán mở trong suốt gần 50 năm, cho đến khi được chỉ ra là giải được trong thời gian đa thức bởi Agrawal, Kayal, và Saxena năm 2002 [3]).

Chú ý rằng Gödel không tổng quát hoá từ $O(n)$ và $O(n^2)$ thành thời gian đa thức. Ông thích thú hơn với các thuật toán có thể chạy tốt trong thực hành. Ông cũng không tính thời gian chạy theo khái niệm hiện đại "độ dài đầu vào" ("input length"). Nếu làm như vậy, ông đã phải chỉ ra một cách cụ thể rằng n được viết theo hệ đơn phân (unary). (Nếu n được viết theo hệ nhị phân (binary) thông thường, thì thời gian tìm kiếm vét cạn trong bài toán của ông có thể theo hàm mũ kép (doubly exponential) của kích thước đầu vào.) Mặt khác, ông hình như có giả thiết kích thước đầu vào được viết theo hệ nhị phân, hoặc ít nhất là thập phân, khi thảo luận về bài toán kiểm tra tính nguyên tố. Thêm vào đó, ông sử dụng ý tưởng đánh giá thời gian chạy của các thuật toán và bài toán trong trường hợp xấu nhất, điều này hoàn toàn không thông thường ở thời điểm đó, nhưng lại là ý tưởng thống trị nền nghiên cứu thuật toán hiện đại. Và hình như ông có ý tưởng về một lớp các bài toán giải được bằng tìm kiếm vét cạn, và đây có thể xem như một tổng quát hoá của NP, và câu hỏi cuối cùng của ông gợi ý đến câu hỏi P và NP. Tóm lại, lá thư của Gödel, khi được phát hiện, đã ngay lập tức được công nhận là một tiền thân quan trọng của lý thuyết NP-đầy đủ. Khi một giải thưởng hàng năm cho các bài báo xuất sắc trong lý thuyết khoa học máy tính được thành lập năm 1992, nó ngay lập tức được đặt tên là giải thưởng Gödel. Gần đây, lá

thư của Gödel thậm chí còn trở thành tên của một blog hay và phổ biến về thuật toán và độ phức tạp tính toán (*Gödel's Lost Letter and P = NP*, http://rjlipton.wordpress.com).

Một nhà toán học nổi tiếng khác với những lá thư dự báo lý thuyết NP-đầy đủ là John Nash. Ông đạt giải thưởng Nobel về kinh tế và đồng thời là chủ đề chính trong cả cuốn sách lẫn bộ phim cùng tên *A Beautiful Mind*. Năm 1955, Nash gửi rất nhiều thư tay về mã hóa đến Cơ quan An ninh quốc gia Hoa Kỳ (United States National Security Agency), những lá thư này không được giải mật và công khai cho đến tận năm 2012 [1]. Trong những lá thư này, ông nhận xét rằng trong các quy trình mã hóa dựa vào khóa (key-based encryption processes) điển hình, nếu các bản rõ (plain texts version) và bản mã hóa (encrypted version) của một số lượng nhỏ các thông tin được cho trước, thì khóa (key) được xác định. Điều này không hoàn toàn đúng về mặt kỹ thuật, bởi vì thêm vào đó cần có đủ entropy trong các bản rõ, nhưng các lập luận của Nash cũng có thể được áp dụng trong bài toán tìm *một số* khóa đồng nhất với quá trình mã hóa. Nhận xét chủ yếu của ông là thậm chí ngay cả khi khóa được xác định, việc tìm kiếm nó có thể không dễ dàng.

Nếu khóa là một chuỗi nhị phân có độ dài r, có thể thực hiện tìm kiếm vét cạn (tương tự như với trường hợp của Gödel), nhưng việc này sẽ cần thời gian theo hàm mũ của r. Đối với các hệ mã hóa yếu, ví dụ như mã hóa thay thế (substitution ciphers), có những kỹ thuật khác nhanh hơn chạy trong thời gian $O(r^2)$ hay $O(r^3)$, nhưng Nash đưa ra giả thuyết rằng "với hầu hết các loại mã hóa đủ phức tạp", thời gian chạy theo hàm mũ của độ dài khóa là không thể tránh khỏi.

Giả thuyết này sẽ suy ra P $\neq$ NP, bởi vì bài toán giải mã ông đề cập đến tương đương theo thời gian đa thức (polynomial-time equivalent) với bài toán sau thuộc lớp NP: Cho trước dữ liệu về bản rõ và bản mã hóa và một tiền tố (prefix) x của một khóa, có tồn tại hay không một khóa đồng nhất với quá trình mã hóa mà có x là một tiền tố? Tuy nhiên, đây là một giả thuyết mạnh hơn bởi vì nó sẽ loại bỏ khả năng toàn bộ các bài toán trong lớp NP có thể, ví dụ như, được giải trong thời gian $n^{O(\log n)}$ – mặc dù không phải thời gian đa thức, nó cũng không phải điều mọi người thường nghĩ tới khi nói về "hàm mũ." Nash cũng đưa ra một khẳng định nhỏ rằng cơ bản là toàn bộ các bài toán giải mã là NP-khó. Khẳng định này có vẻ là sai. Nash đã đưa ra một sơ đồ mã hoá của loại mã hoá mà ông đã đề cập, nhưng NSA nhận xét trong các ghi chú riêng rằng nó chỉ cung cấp sự bảo mật giới hạn, và từ khi các lá thư được công khai, rất nhiều nhà nghiên cứu hiện đại đã chỉ ra rằng nó dễ dàng bị bẻ khoá [2]. Thêm vào đó, giống như Gödel, Nash không tổng quát hoá từ thời gian đa thức bậc thấp lên thời gian đa thức tổng quát. Tuy nhiên, ông đã dự đoán một cách chính xác về sự khó khăn khi giải bài toán P và NP trong toán học. Ông đã thừa nhận rằng ông không thể chứng minh giả thuyết của mình, hay mong đợi rằng nó có thể được chứng minh, thậm chí ngay cả khi nó đúng.

Cook, Karp, và Levin

Lý thuyết NP-đầy đủ về cơ bản bắt nguồn từ bài báo "The complexity of theorem-proving procedures" [14] của Steve Cook năm 1971. Bài báo đưa ra các kết quả đầu tiên về NP-đầy đủ. Tuy nhiên, Leonid Levin, lúc đó là một sinh viên ở Moscow, cũng chứng minh các kết quả gần giống như thế trong khoảng cùng một thời gian, mặc dù các kết quả của ông không được công bố cho đến tận năm 1973. Qua nhiều năm, bản chất đồng thời và độc lập của các kết quả của Levin đã vượt qua sự khác biệt về thời gian công bố, và điều từng được gọi là "Định lý Cook"

Hình 1: Stephen Cook, Richard Karp, và Leonid Levin, các bức ảnh từ những năm 1980

ngày nay thường được nhắc đến như là "Định lý Cook-Levin." Tôi sẽ nói một chút về những sự phát triển song song này.

Khi Cook viết bài báo của mình, ông là một Phó giáo sư (Associate Professor) ở Khoa Khoa học máy tính của Đại học Toronto, nơi mà hiện nay ông là Giáo sư (University Professor). Trước đó, ông nhận bằng Tiến sĩ (PhD) từ Harvard năm 1966, và dành bốn năm làm Trợ lý giáo sư (Assistant Professor) ở Đại học California, Berkeley, nơi đã từ chối một cách ngu ngốc việc bổ nhiệm chính thức (tenure) của ông. Bài báo của Cook xuất hiện ở tập san của hội thảo ACM Symposium on Theory of Computing (STOC) năm 1971, và có những câu chuyện rỉ tai rằng nó suýt nữa thì không được chấp nhận. Điều này dường như không thể, mặc dù không phải là lần đầu tiên một kết quả mang tính đột phá không được công nhận khi nó xuất hiện. Tầm quan trọng của bài báo gần như được công nhận ngay khi nó xuất hiện. Bài báo không chỉ chứng minh bài toán SASTISFIABILITY là NP-đầy đủ (theo ngôn ngữ hiện đại) mà còn chỉ ra kết quả tương tự cho bài toán 3SAT và gợi ý về những ứng dụng rộng lớn hơn của khái niệm NP-đầy đủ thông qua việc chứng minh kết quả tương tự cho bài toán SUBGRAPH ISOMORPHISM (cụ thể hơn là một trường hợp riêng của bài toán mà ngày nay được biết đến như là bài toán CLIQUE). Lúc đó tôi là một sinh viên cao học ở MIT, và Albert Meyer và Mike Fischer đã đưa những kết quả này vào giảng dạy trong khoá học về Thuật toán của họ mùa thu năm 1971. Những người khác cũng đang bận rộn, và điều này trở nên rõ ràng hơn ở hội thảo "Complexity of Computer Computations" tổ chức tháng 03 năm 1972 ở Trung tâm nghiên cứu IBM T.J. Watson ở Yorktown Heights, NY, khi Richard Karp trình bày bài báo nổi tiếng của mình.

Karp cũng nhận bằng Tiến sĩ từ Harvard (1959), và sau 11 năm gắn bó với Trung tâm nghiên cứu IBM nơi tổ chức hội thảo, ông đã chuyển sang làm Giáo sư ở UC Berkeley năm 1968 và ở lại đó cho đến tận ngày nay, sau một cuộc tạm trú ngắn ở Đại học Washington ở Seattle. Bài báo của Karp chỉ ra 19 bài toán NP-đầy đủ khác, trong đó bao gồm các bài toán nổi tiếng ngày nay như VERTEX COVER, CHROMATIC NUMBER, cả phiên bản có hướng và vô hướng của HAMILTONIAN CIRCUIT, SUBSET SUM, và KNAPSACK. Hầu hết các chứng minh được hoàn thành bởi Karp, nhưng một vài kết quả được quy cho Gene Lawler, Bob Tarjan, và "các seminar về thuật toán ở Cornell." Bài báo của ông dường như là bài báo đầu tiên sử dụng các ký hiệu P và NP, mặc dù "NP-đầy đủ" lúc đó được gọi là "đa thức đầy đủ" ("polynomial complete"), một thuật ngữ được sử dụng trong rất nhiều các bài báo trước đó, trước khi thuật ngữ hiện đại được sử dụng. Bài báo cũng giới thiệu sự khác nhau giữa một *phép chuyển trong thời gian đa thức*

(*polynomial transformation*), trong đó một trường hợp của bài toán thứ nhất được chuyển thành một trường hợp của bài toán thứ hai và chúng có cùng kết quả đúng-sai, và một *phép quy về trong thời gian đa thức* (*polynomial reduction*), trong đó bài toán đầu tiên được giải bằng cách sử dụng một hay nhiều lần gọi một hàm con (subroutine) giải bài toán thứ hai. Cook phát biểu các kết quả của ông theo khái niệm thứ hai, nhưng các chứng minh của ông gần như là dựa vào khái niệm thứ nhất.

Đó là hội thảo đầu tiên tôi tham gia, và tôi đã khá ấn tượng với tất cả những người tham dự nổi tiếng mà tôi lần đầu được gặp - bao gồm John Hopcroft, Michael Rabin, Bob Tarjan, Jeff Ullman, và cả bản thân Richard Karp. Tôi thậm chí còn ngồi đối diện Dick trong một buổi ăn trưa. Tôi tận dụng cơ hội đó để đề cập với ông rằng tôi đã vừa tự chứng minh một kết quả đa thức đầy đủ cho bài toán BIN PACKING, bài toán chủ đề cho khoá luận của tôi. Albert Meyer đã đề nghi tôi làm việc với bài toán này chỉ một tháng trước đó, và nói với tôi "Điều này là hoàn hảo cho cậu, Johnson. Cậu không cần biết cái gì cả – cậu chỉ cần thông minh." Albert biết về bài toán từ một bản tiền ấn phẩm của bài báo ở STOC 1972 viết bởi Garey, Graham, và Ullman [21]. Trong bài toán, bạn được cho trước một dãy các số $a_1, a_2, \ldots, a_n \in (0, 1]$ và một số k, và câu hỏi là liệu các số đã cho có thể được chia thành k tập, trong đó mỗi tập có tổng các số không vượt quá 1. Dick tỏ ra thích thú, nhưng, ngay khi vừa nói xong, tôi đã cảm thấy xấu hổ khi nhận ra rằng chứng minh của tôi hiển nhiên đến mức nào khi so sánh với chứng minh trong bài báo của ông (SUBSET SUM là một trường hợp đặc biệt của BIN PACKING khi $k = 2$ và $\sum_{i=1}^{n} a_i = 2$.)

Bên cạnh rất nhiều bài báo thú vị, hội thảo cũng bao gồm một buổi thảo luận nhóm trực tiếp (panel discussion) và một bản tóm tắt những bài trong tập san [46]. Cuộc thảo luận cũng đề cập đến các vấn đề từ những lần trình bày trước, nhưng luôn luôn quay lại câu hỏi về P và NP. Nhận xét đáng nhớ (và tiên tri) nhất từ cuộc thảo luận là của John Hopcroft. Ông nhận thấy rằng, mặc dù có vẻ như có một sự nhất trí dần được hình thành là hai lớp này không bằng nhau, với tất cả những gì chúng ta đã biết, mọi bài toán thuộc NP có lẽ có thể được giải trong thời gian tuyến tính. Ông kết luận rằng sẽ là "an toàn một cách hợp lý" khi giả thiết rằng, trong vòng năm năm tới, không ai sẽ chứng minh rằng bất kỳ bài toán đa thức đầy đủ nào thậm chí yêu cầu nhiều hơn thời gian bậc hai. Hiện tại sau 40 năm và còn tiếp tục, và chúng ta vẫn chưa thấy bất kỳ chứng minh nào như thế cả.

Cùng lúc đó, ở một thế giới hoàn toàn khác, Leonid Levin cũng nghĩ tới các vấn đề này, nhưng các kết quả của ông không nhận được mức độ quan tâm tương tự từ cộng đồng. Ở Liên Xô thời điểm đó, rất nhiều nhà nghiên cứu quan tâm đến các vấn đề liên quan đến câu hỏi P và NP. Đặc biệt, có một khái niệm về một lớp các bài toán mà chỉ có thể giải bằng *perebor*, một danh từ tiếng Nga cho các thuật toán gần như dựa trên tìm kiếm vét cạn [52]. Levin đã từng là nghiên cứu sinh tiến sĩ ở Đại học Moscow. Năm 1971, ông hoàn thành một khoá luận về độ phức tạp Kolmogorov, nhưng mặc dù nó được chấp nhận bởi Kolmogorov (giáo sư hướng dẫn của ông) và một hội đồng chấm khoá luận, nhà chức trách đã từ chối cấp bằng do các lý do về chính trị. (Levin thừa nhận rằng bản thân có đôi chút bất trị trong việc tuân thủ các quy tắc ở Liên Xô [51, 151–152].) Levin tiếp tục làm việc với những vấn đề khác, tuy nhiên, đặc biệt là *perebor*, và đưa ra phiên bản NP-đầy đủ của chính ông trong cùng năm đó, và trình bày về nó ở rất nhiều seminar khác nhau ở Moscow và Leningrad [52]. Ông cũng viết về các kết quả của mình và nộp để công bố vào tháng 06 năm 1972 [52], mặc dù bài báo không xuất hiện cho đến nửa sau của 1973. Tiêu đề của bài báo, dịch sang tiếng Anh, là "Universal sequential search problems" [44] ("Sequential search" là một cách dịch sai của *perebor*).

Bài báo hai trang ngắn gọn và súc tích, một đặc điểm được chia sẻ bởi rất nhiều các bài báo của Levin sau đó (ví dụ, xem [55, 45]), hoàn toàn bỏ qua các chứng minh. Một bản dịch có chỉnh sửa xuất hiện ở phần phụ lục của [52]. Trong bài báo của mình, Levin xét sự tổng quát hoá của NP thành các bài toán tìm kiếm: Các quan hệ (relation) $A(x, y)$ trên các chuỗi ký tự sao cho với tất cả các cặp (x, y) thoả mãn điều kiện $A(x, y)$ đúng, độ dài của y bị chặn bởi một hàm đa thức của độ dài của x, và sao cho với mọi cặp (x, y), có thể xác định trong thời gian đa thức liệu $A(x, y)$ có đúng hay không. Ở đây x đại diện cho một trường hợp của bài toán, và y đại diện cho "lời giải" tương ứng. Bài toán tìm kiếm cho A là bài toán cho x, tìm y sao cho $A(x, y)$ đúng. Bài toán tương ứng trong NP là bài toán cho x, liệu có tồn tại y sao cho $A(x, y)$ đúng. Levin đề cập phiên bản này, gọi nó là một bài toán "gần như tìm kiếm" ("quasi-search" problem), nhưng tập trung vào phiên bản bài toán tìm kiếm. Ông mô tả thứ mà ngày nay chúng ta nhận biết như là khái niệm cơ bản của phép quy về trong thời gian đa thức (polynomial reduction) từ một bài toán tìm kiếm A sang một bài toán khác, và gọi một bài toán tìm kiếm là "bài toán tìm kiếm phổ quát" ("universal search problem") nếu tồn tại các phép quy về trong thời gian đa thức đến nó từ tất cả các bài toán tìm kiếm trong lớp trên. Ông tiếp đó liệt kê sáu bài toán tìm kiếm mà ông có thể chứng minh chúng là các bài toán tìm kiếm phổ quát. Những bài toán này bao gồm phiên bản tìm kiếm của SATISFIABILITY, SET COVER, và SUBGRAPH ISOMORPHISM, cùng với các bài toán khác không có trong danh sách của Karp, ví dụ như bài toán lát gạch sau: cho một lưới hình vuông có các ô nằm ở biên thoả mãn điều kiện mỗi ô có một số trong khoảng từ 1 đến 100, cùng với các quy tắc hạn chế nội dung của các ô bên trong khi biết trước nội dung của bốn ô hàng xóm (ở bên trái, phải, trên, và dưới), tìm một cách xếp các ô đúng luật tương ứng với các ô biên cho trước.

Những người đã nghe Levin nói về các kết quả này ngay lập tức bị ấn tượng. Trakhtenbrot [52] trích lời của Barzdin, người đã nghe Levin nói ở Novosibirsk tháng 04 năm 1972, rằng "Mới đây thôi Levin vừa nói cho tôi các kết quả mới của anh; đó là một bước ngoặt trong chủ đề về *perebor*!". Chú ý rằng đây là một bằng chứng hiển nhiên về việc các kết quả của Cook và Karp chưa hề nhận được sự chú ý rộng rãi ở Nga. Tuy nhiên, các kết quả của Levin cũng vậy. Năm 1973, khi các nhà nghiên cứu lý thuyết ở Nga cuối cùng tiếp nhận NP-đầy đủ, điều này được thực hiện chủ yếu qua các bài báo của Cook và Karp [25]. Ảnh hưởng của Levin dường như không quá trở nên rộng khắp ngoại trừ đối với những người đã từng trực tiếp nghe ông trình bày.

Năm 1978, Levin nhập cư vào Mỹ, nơi tôi gặp ông lần đầu tiên khi đến thăm MIT. Ở đây, ông cuối cùng nhận được một bằng Tiến sĩ chính thức năm 1979, sau đó ông nhận một vị trí ở Đại học Boston, nơi hiện tại ông là Giáo sư (Full Professor). Ông đã đóng góp thêm rất nhiều vào lý thuyết độ phức tạp tính toán, bao gồm

- Một lý thuyết về tính đầy đủ trong trường hợp trung bình (average case completeness) [45], sử dụng thứ mà ông chỉ ra rằng một phiên bản của bài toán lát gạch nêu trên của ông, với một khái niệm tự nhiên về một phân phối đều (uniform distribution) cho nó, có thể được giải trong thời gian kỳ vọng đa thức (polynomial expected time) trừ khi mọi tổ hợp khác của một bài toán trong NP với một phân phối xác suất hợp lý cũng có thể được giải trong thời gian như vậy.

- Một chứng minh rằng tồn tại các hàm một chiều (one-way functions) cần thiết cho mật mã (cryptography) khi và chỉ khi các bộ sinh số giả ngẫu nhiên (pseudorandom number generators) tồn tại mà không thể phân biệt chúng với các bộ sinh số ngẫu nhiên thật trong thời gian đa thức [28].

- Một chứng minh rằng một thuật toán tiền thân của thuật toán ellipsoid công bố năm 1965, trong đó các đơn hình (simplex) đóng vai trò của các hình elip, cũng chạy trong thời gian đa thức [55] (do đó, *có* một thuật toán đơn hình (simplex algorithm) chạy trong thời gian đa thức ...).

Cook và Karp cũng có nhiều đóng góp quan trọng cho lý thuyết độ phức tạp tính toán kể từ khi công bố các đột phá của mình. Nhiều đóng góp của Karp được biết đến rộng khắp trong cộng đồng quy hoạch toán học (mathematical programming) và quá rộng để có thể liệt kê ở đây. Đóng góp chính của Cook chủ yếu ở việc nghiên cứu độ phức tạp chứng minh (proof complexity), tuy nhiên, ông cũng đóng góp trong việc giới thiệu ít nhất một lớp độ phức tạp (complexity class) mới, và lớp này cung cấp một góc nhìn thú vị về NP-đầy đủ.

Đó là lớp SC, một lớp bao gồm các bài toán quyết định (decision problems) có thể giải được bằng các thuật toán chạy trong thời gian đa thức và chỉ yêu cầu không gian polylogarithmic, nghĩa là, sử dụng $O(\log^k n)$ không gian với một số k cố định nào đó. Ở đây "SC" là viết tắt của "Steve's Class," và tên gọi này được đề nghị bởi Nick Pippenger nhằm vinh danh một kết quả kinh ngạc của Steve năm 1979 là các ngôn ngữ phi ngữ cảnh tất định (deterministic context-free languages) đều thuộc lớp này [15], và đồng thời cũng là để trả đũa cho việc Steve giới thiệu thuật ngữ "NC" ("Nick's Class") để mô tả lớp các bài toán quyết định giải được trong thời gian polylogarithmic và chỉ yêu cầu một số lượng đa thức các bộ xử lý song song (parallel processors) [26]. Điểm quan trọng của hai lớp này là ở chỗ mặc dù dễ thấy rằng chúng đều nằm trong P, ta có thể hi vọng rằng chúng đều là các lớp con *thực sự* (proper subclasses) của P. Nghĩa là, có lẽ tồn tại các bài toán trong P không thể giải được trong thời gian đa thức nếu chỉ cho phép sử dụng không gian polylog, và các bài toán không thể giải được trong thời gian polylog nếu chỉ cho phép sử dụng một số lượng đa thức các bộ xử lý song song. Tương tự như với NP-đầy đủ, ta có thể xác định các ứng cử viên cho những bài toán này bằng cách xác định những bài toán là "đầy đủ cho P" với các phép quy về phù hợp. Một ví dụ nổi tiếng về bài toán đầy đủ cho P theo cả hai nghĩa trên là bài toán LINEAR PROGRAMMING [16].

Cả Cook và Karp đều nhận được rất nhiều giải thưởng. Cook được nhận giải thưởng Turing (giải thưởng hàng đầu trong ngành khoa học máy tính) năm 1982 và giải thưởng CRM-Fields năm 1999 (giải thưởng hàng đầu ở Canada cho những đóng góp về mặt nghiên cứu trong các ngành toán học). Karp nhận giải thưởng Lanchester năm 1977, giải thưởng Fulkerson trong lĩnh vực toán rời rạc năm 1979, giải thưởng Turing năm 1985, giải thưởng ORSATIMS von Neumann Theory năm 1990, và nhiều giải thưởng khác. Levin thì quá hạn từ lâu cho giải thưởng lớn của chính bản thân ông, mặc dù tôi mong đợi điều này sẽ đến sớm. Và tất nhiên, giải thưởng lớn nhất liên quan đến NP-đầy đủ vẫn chưa được trao: câu hỏi liệu P có bằng NP là một trong sáu bài toán mở còn lại mà người giải sẽ được Viện toán học Clay trao thưởng một triệu đô la Mỹ.

Garey, Johnson, và Computers and Intractability

Mối liên hệ có tầm ảnh hưởng nhất của tôi đến lý thuyết NP-đầy đủ không thể nghi ngờ là cuốn sách *Computers and Intractability: A Guide to the Theory of NP-completeness* tôi viết cùng Mike Garey xuất bản năm 1979. Ở thời điểm đó, chúng tôi tự hào hứa với các nhà xuất bản rằng

Hình 2: Michael Garey và David Johnson năm 1977

chúng tôi sẽ bán 5000 bản, nhưng cho đến nay cuốn sách đã được in 50000 bản cùng với khoảng 40000 lượt trích dẫn theo Google Scholar.

Những liên hệ đầu tiên của tôi với lý thuyết này, ngoại trừ cuộc nói chuyện trong bữa trưa đề cập ở trên, chủ yếu liên quan đến một trong các phương pháp xử lý các bài toán NP-đầy đủ: thiết kế và đánh giá các thuật toán xấp xỉ. Trong khi ở MIT, tôi viết một luận án Tiến sĩ về các thuật toán xấp xỉ cho bài toán đóng thùng (bin packing) [32] và một bài báo đề xuất việc làm thế nào hướng tiếp cận giống như thế có thể được mở rộng sang các bài toán khác, ví dụ như tô màu đồ thị (graph coloring), phủ tập hợp (set covering), và thoả mãn cực đại (maximum satifiability) [33].

Nhờ vào sự mạnh mẽ của nghiên cứu này, tôi được Ron Graham và Mike Garey, những người mà bài báo ban đầu của họ về bài toán đóng thùng đã đưa tôi đến với lý thuyết NP-đầy đủ, nhận vào làm việc ở Bell Labs. Sau khi nhận bằng Tiến sĩ vào tháng 06 năm 1973, tôi chuyển đến New Jersey và bắt đầu sự nghiệp Bell Labs/AT&T của mình. Một trong số những hợp tác đầu tiên của tôi và Mike là trong việc đưa ra phản hồi cho một lá thư mà Don Knuth đã viết vào tháng 10 cho rất nhiều chuyên gia trong ngành. Lá thư đó tìm kiếm một tên gọi tốt hơn là "đa thức đầy đủ" cho lớp các bài toán mà Cook và Karp đã xác định. Knuth yêu cầu một cuộc bỏ phiếu cho ba cụm từ mà ông đã đề xuất ("Herculean," "formidable," và "arduous"). Chúng tôi không đặc biệt thích bất kỳ cụm từ nào Knuth đề xuất, và chúng tôi đã đề nghị thêm cụm từ "NP-đầy đủ" như là một ứng viên. Chúng tôi không phải là những người duy nhất, và khi Knuth thông báo về cuộc thăm dò của ông tháng 01 năm 1974 [43], ông từ bỏ các đề nghị ban đầu của mình, và công bố "NP-đầy đủ" chiến thắng, với "NP-khó" được chọn để chỉ các bài toán ít nhất khó ngang với tất cả các bài toán trong NP, mặc dù bản thân chúng có thể không thuộc NP. Một tóm tắt thú vị về các đề xuất khác mà Knuth nhận được có trong bài viết của ông hoặc [23].

Mike và tôi cũng bắt đầu một quá trình hợp tác năng động về các thuật toán cho cả bài toán đóng thùng (bin packing) và bài toán lập kế hoạch (scheduling), cùng với việc chứng minh các kết quả NP-đầy đủ mới. Khi Karp viết một bài báo tạp chí [40] từ bài báo đăng trên tập san hội thảo ban đầu, ông mở rộng danh sách các bài toán NP-đầy đủ lên 25 bài, thêm vào một số các kết quả mới. Bài báo này đặt cơ sở cho cuốn sách của chúng tôi [23], với một danh sách dài hơn, mặc dù nguồn gốc thực sự của cuốn sách phần nhiều là do trùng hợp. Vào tháng 04 năm 1976, Mike và tôi tham dự một cuộc hội thảo ở Đại học Carnegie-Mellon về "những hướng mới và kết quả gần đây về thuật toán và độ phức tạp tính toán," ("New Directions and Recent Results in Algorithms and Complexity,") nơi tôi trình bày về các loại kết quả xấp xỉ khác nhau mà chúng

tôi đã thấy từ trước đến nay. Sau đó, trong một buổi nghỉ giữa giờ, một biên tập viên của công ty xuất bản Prentice-Hall bắt chuyện với tôi và đề nghị Mike và tôi viết một cuốn sách về các thuật toán xấp xỉ. Khi nghĩ về đề nghị này, chúng tôi nhận ra một thứ cần thiết trước khi có bất kỳ cuốn sách nào về các thuật toán xấp xỉ là một cuốn sách về NP-đầy đủ, và khi rời khỏi hội thảo, chúng tôi đã ở ngay trên con đường quyết định tự viết cuốn sách đó.

Một trong số các nhiệm vụ của tôi là thu thập các kết quả NP-đầy đủ cho danh sách dự định của chúng tôi, mà trong những ngày đó khi máy tính cá nhân còn chưa xuất hiện, nghĩa là viết các chi tiết bằng tay trên các tấm thẻ lưu trữ trong hộp nhựa. Ở thời điểm đó, việc đặt mục tiêu xét toàn bộ các kết quả vẫn là có thể, và danh sách cuối cùng của chúng tôi gồm khoảng 300 bài toán bao trùm hầu hết các kết quả đã được công bố cho đến khi chúng tôi hoàn thành bản nháp đầu tiên giữa năm 1978, gồm cả rất nhiều kết quả chúng tôi tự chứng minh khi xác định các khoảng trống thú vị trong hoàn cảnh nghiên cứu hiện tại, và với chúng, chúng tôi cung cấp một trích dẫn vô ích "[Garey and Johnson, unpublished]." Chúng tôi vẫn giữ ghi chép về các chứng minh (trong các hộp nhựa tương tự) và phần lớn trong số đó vẫn có thể được xây dựng lại ... Sau các cuộc thảo luận chi tiết về những gì chúng tôi muốn nói, tôi viết các bản nháp đầu tiên của các chương, và Mike sau đó làm rõ hơn và cải thiện phần viết lách. (Một so sánh nhanh giữa những gì viết trong [23] với những gì viết ở bài viết này sẽ có thể sẽ khiến hầu hết các độc giả ước gì Mike vẫn tiếp tục làm công việc đó.)

Chúng tôi có sử dụng máy tính khi thực sự tiến hành sắp chữ cho cuốn sách, mặc dù tôi đã phải lê lết lên tận phòng UNIX ở tầng 5 để gõ văn bản, và phải chịu đựng mùi hoá chất nồng nặc từ các máy sắp chữ quang cổ lỗ ở đó. Vì chúng tôi cung cấp bản sao hoàn tất (camera-ready copy), chúng tôi có quyền quyết định mọi thứ nhìn như thế nào, mặc dù nhà xuất bản cũng cung cấp các nhận xét kỹ lưỡng và hữu ích trong việc sao chép và chỉnh sửa, bao gồm cả việc dạy chúng tôi một lần và mãi mãi về tất cả những sự khác biệt giữa "that" và "which." Chỉ có một trục trặc nhỏ ở phút cuối cùng rất may mắn được phát hiện trước khi cuốn sách chính thức được in – bìa cuốn sách dự định mô tả kết quả của tích đồ thị (graph product) của một tam giác (triangle) và một đường thẳng (path) có độ dài bằng hai, và lần vẽ đầu tiên của đồ thị này thiếu rất nhiều cạnh.

Qua nhiều năm, cuốn sách gần như không thay đổi, mặc dù trong các lần in sau, hai trang của mục "Cập nhật" được thêm vào cuối cùng để liệt kê danh sách các lỗi in và thông báo về tình trạng của mười hai bài toán mở liệt kê ở Phụ lục A13 của cuốn sách. Tính đến ngày nay, chỉ hai bài toán trong số chúng vẫn chưa có lời giải: GRAPH ISOMORPHISM và PRECEDENCE CONSTRAINED 3-PROCESSOR SCHEDULING. Trong số mười bài còn lại, năm bài hiện đã biết rằng giải được trong thời gian đa thức, và năm bài khác thuộc NP-đầy đủ. Để biết thêm chi tiết, xem [35, 39]. Việc tái bản cuốn sách lần hai vẫn luôn là dự định nhưng chưa bao giờ được bắt đầu, mặc dù tôi đã tiếp tục viết cột về NP-đầy đủ của mình, hiện nay xuất hiện lẻ tẻ trong tạp chí *ACM Transactions on Algorithms*, để xây dựng một cơ sở cho việc tái bản cuốn sách.

Chúng tôi chưa bao giờ viết cuốn sách về các thuật toán xấp xỉ đó, và trên thực tế không có cuốn sách nào như thế xuất hiện cho đến khi *Approximation Algorithms for NP-Hard Problems* [29] của Dorit Hochbaum xuất bản năm 1997. Đây là một tập hợp các bài viết có chỉnh sửa, và trong đó Mike, Ed Coffman, và tôi có đóng góp một chương. Cuốn sách giáo khoa đầu tiên về các thuật toán xấp xỉ là *Approximation Algorithms* [53] của Vijay Vazirani, và nó mãi đến tận năm 2001 mới xuất hiện. Mặc dù Mike và tôi chưa bao giờ viết cuốn sách tái bản, trên thực tế, theo nghĩa nào đó, còn có một cuốn "Garey and Johnson" thứ hai. Năm 1990, những người vợ của

chúng tôi, Jenene Garey và Dorothy Wilson, tương ứng là một Giáo sư về dinh dưỡng ở NYU và một giáo viên phổ thông, đồng tác giả cuốn sách *The Whole Kid's Cookbook*, và các bản in của cuốn sách được bán để gây quỹ cho Trung tâm chăm sóc trẻ em đỉnh cao (Summit Child Care Center), một trung tâm ở địa phương nơi mà Dorothy đã từng làm việc.

Bốn Mươi Năm: Sự Khó Khăn Của Xấp Xỉ

Việc trình bày một cách toàn vẹn về lịch sử phát triển của lý thuyết NP-đầy đủ từ những năm 1907 sẽ là điều không thể trong giới hạn bài viết này. Do đó, trong mục này, tôi sẽ chỉ tập trung vào một vấn đề: ứng dụng của lý thuyết trong các thuật toán xấp xỉ.

Một thuật toán xấp xỉ không nhất thiết phải trả lại một lời giải tối ưu (optimal solution), nhưng nó thường hướng đến một lời giải khả thi (feasible solution) nào đó mà mọi người hi vọng là sẽ gần tối ưu (near-optimal). Một cách điển hình để đánh giá một thuật toán xấp xỉ A là dựa trên "đảm bảo trong trường hợp xấu nhất" ("worst-case guarantee") mà nó cung cấp. Chúng ta hãy giả sử một cách đơn giản rằng bài toán X mà thuật toán A cần xấp xỉ là một bài toán cực tiểu hoá (minimization problem). Từ đó, A cung cấp một đảm bảo trong trường hợp xấu nhất bằng với giá trị lớn nhất, trên toàn bộ mọi trường hợp I của bài toán, của $A(I)/OPT(I)$, với $A(I)$ là giá trị của lời giải mà thuật toán đưa ra cho trường hợp I, và $OPT(I)$ là giá trị của lời giải tối ưu. Ví dụ, thuật toán Christofides cho bài toán người bán hàng (Traveling Salesman Problem – TSP) có đảm bảo trong trường hợp xấu nhất là $3/2$ nếu chúng ta chỉ quan tâm đến các trường hợp của bài toán thoả mãn bất đẳng thức tam giác [12].

Tất nhiên là chúng ta thích thú hơn với các thuật toán xấp xỉ trong thời gian đa thức cho các bài toán NP-khó. Thật không may, trên thực tế đôi khi việc thiết kế một thuật toán như vậy có thể khó ngang với việc tìm kiếm một lời giải tối ưu. Bài báo đầu tiên đưa ra nhận xét này được viết bởi Sahni và Gonzalez [49] năm 1974. Họ đã chỉ ra rằng, ví dụ như, nếu *không* có giả thiết bất đẳng thức tam giác, thì với bất kỳ hằng số k nào, sự tồn tại của một thuật toán xấp xỉ chạy trong thời gian đa thức cho bài toán TSP với đảm bảo trong trường hợp xấu nhất bằng k hoặc tốt hơn sẽ suy ra P $=$ NP. Chứng minh của họ liên quan tới một cách xây dựng "lỗ hổng" ("gap" construction), bằng cách chuyển các trường hợp của HAMILTON CIRCUIT sang các trường hợp của TSP trong đó các chu trình tối ưu (optimal tours) có độ dài bằng n nếu chu trình Hamilton tồi tại, và có độ dài lớn hơn kn nếu nó không tồn tại (ví dụ có thể đặt khoảng cách giữa u và v bằng 1 nếu $\{u, v\}$ là một cạnh trong đồ thị ban đầu, và bằng kn trong trường hợp còn lại).

Tính đến khi cuốn sách về NP-đầy đủ của chúng tôi xuất hiện, đã có thêm một vài kết quả thú vị thuộc dạng này. Trong đó, các kết quả đặc biệt thú vị liên quan đến việc loại trừ các "sơ đồ xấp xỉ." ("approximation schemes.") Một *sơ đồ xấp xỉ trong thời gian đa thức* (polynomial-time approximation scheme – PTAS) cho một bài toán là một tập hợp các thuật toán xấp xỉ A_ϵ chạy trong thời gian đa thức, trong đó A_ϵ có đảm bảo trong trường hợp xấu nhất là $1 + \epsilon$ hoặc tốt hơn. Năm 1975, Sahni [48] đã chỉ ra rằng bài toán xếp ba lô (Knapsack Problem) có một sơ đồ như thế. Các thuật toán của ông, và rất nhiều thuật toán tương tự, đều không thể chạy trong thực hành, và có thời gian chạy theo hàm mũ của $1/\epsilon$, mặc dù với ϵ cố định bất kỳ chúng đều chạy trong thời gian đa thức. Tuy nhiên, qua nhiều năm, vẫn có rất nhiều nỗ lực trong việc tìm kiếm các sơ đồ như thế cho một lượng lớn các bài toán.

Khi đã biết về việc các PTAS khó có thể được ứng dụng trong thực hành như thế nào, ta có thể xem việc thiết kế chúng như là việc đưa ra các kết quả "tiêu cực-tiêu cực", hơn là các kết quả tích cực. Ta cũng có thể loại bỏ sự tồn tại của sơ đồ dạng này cho một bài toán (giả thiết rằng P $\neq$ NP) bằng cách chứng minh sự tồn tại của một ϵ sao cho không có thuật toán xấp xỉ nào chạy trong thời gian đa thức mà có đảm bảo trong trường hợp xấu nhất là $1+\epsilon$ hoặc tốt hơn trừ khi P = NP. Điều này hiển nhiên đúng với BIN PACKING, bởi vì nếu một thuật toán có thể đảm bảo một tỷ lệ nhỏ hơn $3/2$ thì ta có thể dùng nó cho bài toán SUBSET SUM. Chứng minh sự tồn tại của một PTAS cho một bài toán do đó đơn thuần là việc chỉ ra rằng không tồn tại ϵ sao cho ta có thể chứng minh rằng không có thuật toán xấp xỉ nào chạy trong thời gian đa thức mà có đảm bảo trong trường hợp xấu nhất là $1+\epsilon$ hoặc tốt hơn.

Tuy nhiên, có một loại PTAS đặc biệt có lẽ có thể được xem xét dưới góc độ tích cực hơn. Ngay sau sự xuất hiện của kết quả về PTAS của Sahni cho bài toán KNAPSACK, Ibarra và Kim [31] đã cải tiến nó một cách đáng kể thông qua việc thiết kế thứ mà ngày nay chúng ta gọi là một sơ đồ xấp xỉ trong thời gian đa thức *hoàn toàn* (*fully* polynomial-time approximation scheme - FPTAS): Một thuật toán A với đầu vào là cả trường hợp I của bài toán lẫn một $\epsilon > 0$, và trả lại một lời giải không tệ hơn $(1+\epsilon)OPT(I)$, và chạy trong thời gian chặn bởi một đa thức không những chỉ của kích thước của I mà còn của cả $1/\epsilon$.

Thật không may, mọi người nhanh chóng nhận ra rằng các FPTAS còn hiếm hơn là PTAS thông thường. Đặc biệt, bài toán TSP với giả thiết bất đẳng thức tam giác không thể có FPTAS trừ khi P $\neq$ NP, một điều không thể bỏ qua với các PTAS thông thường. Điều này là bởi vì bài toán là "NP-khó theo nghĩa mạnh," nghĩa là nó là NP-khó thậm chí ngay cả khi chúng ta hạn chế tất cả các số trong đầu vào (trong trường hợp này là khoảng cách giữa các thành phố) chỉ là các số nguyên bị chặn bởi một đa thức cố định nào đó của độ dài đầu vào, thay vì các giá trị lớn theo hàm mũ thông thường được cho phép bởi ký hiệu nhị phân. Dễ thấy rằng [22] không có bài toán tối ưu NP-khó mạnh nào có thể có FPTAS trừ khi P = NP (nếu điều này xảy ra thì chả cần FPTAS nào).

Ở đầu kia của cán cân (các bài toán mà không có thuật toán nào có một đảm bảo hữu hạn tồn tại, hay ít nhất là được biết đến), chỉ có một vài kết quả, mặc dù đảm bảo tốt nhất có thể cho bài toán SET COVER là $H(n) = \sum_{i=1}^{\infty} 1/i \sim \ln n$ [33, 38], và không có thuật toán nào cho CLIQUE được biết đến với đảm bảo tốt hơn $O(n/polylog(n))$ [33]. Không biết đây đã phải là kết quả tốt nhất có thể (giả thiết P $\neq$ NP) hay không, và cả lĩnh vực nằm trong trạng thái thiếu hiểu biết như vậy trong hơn một thập kỷ. Trên thực tế, mặc dù thi thoảng có một số kết quả thú vị cho bài toán cụ thể, các thuật toán xấp xỉ vẫn chỉ là những chủ đề nhỏ trong nghiên cứu thuật toán cho đến năm 1991, khi một kết quả dường như không liên quan gì trong lý thuyết NP-đầy đủ đột nhiên mang đến cho chúng một cuộc sống mới bùng nổ.

Kết quả đó là sự khám phá về một đặc trưng mới của NP theo "các chứng minh kiểm tra được bằng xác suất" ("probabilistically checkable proofs" – PCPs). Một PCP là một chứng minh mà việc kiểm tra nó có thể được đánh giá bằng cách nhìn vào một vài bit được chọn ngẫu nhiên. Nếu chứng minh là hợp lệ (valid) thì bất kỳ lựa chọn nào của các bit đều hỗ trợ sự thật đó. Nếu nó có tỳ vết (defective) thì một lựa chọn ngẫu nhiên của các bit để đánh giá sẽ, với xác suất $1/2$ hoặc lớn hơn, xác nhận rằng chứng minh là không hợp lệ. Khái niệm cơ sở này dẫn đến sự xuất hiện của một loạt các bài báo, bắt đầu với sự nghiên cứu các chứng minh tương tác (interactive proofs) liên quan đến nhiều người chứng minh (prover) và một người xác minh (verifier). (Trong số những bài báo này có một bài mà Leonid Levin là đồng tác giả [10].)

Cho $f(n)$ và $g(n)$ là hai hàm từ tập các số tự nhiên đến chính nó. Ký hiệu PCP(f, g) là lớp tất cả các bài toán có PCP sử dụng $O(f(n))$ bit ngẫu nhiên và nhìn vào $O(g(n))$ bit của chứng minh. Vào cuối năm 1991, Feige, Goldwasser, Lovász, Safra, và Szegedy [20] chứng minh rằng NP $\subseteq$ PCP$(\log n \log\log n, \log n \log\log n)$, và, đáng ngạc nhiên là, kết quả mang đậm tính kỹ thuật này suy ra rằng CLIQUE không thể được xấp xỉ đến bất kỳ thừa số hằng số nào trừ khi NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$. Đây là một kết luận yếu hơn P = NP, nhưng dễ dàng tin tưởng hơn, và điều kiện của suy luận này đã được cải tiến mạnh hơn thành P = NP vào đầu năm 1992, khi Arora và Safra [7] chứng minh rằng NP = PCP$(\log n, \log n)$. Một khoảng thời gian ngắn sau đó, Arora, Lund, Motwani, Sudan, và Szegedy [5] cải tiến kết quả với NP = PCP$(\log n, 1)$, và điều này thậm chí có các hệ quả mạnh hơn đối với các bài toán xấp xỉ. Đặc biệt, nó suy ra rằng rất nhiều bài toán nổi tiếng, bao gồm MAX 2-SAT, VERTEX COVER, và bài toán TSP với bất đẳng thức tam giác, không thể có PTAS. Bài viết này không có đủ chỗ để miêu tả các chi tiết cụ thể của những chứng minh của các kết quả trên hay toàn bộ các tài liệu tham khảo, nhưng ý tưởng then chốt là việc đưa ra một cách xây dựng lỗ hổng (gap construction) cho bài toán đang xét dựa trên mối liên hệ giữa các bit ngẫu nhiên sử dụng bởi người xác minh trong một PCP của 3-SAT, và các bit sử dụng trong chứng minh ở các địa chỉ xác định bởi những bit ngẫu nhiên này. Bài báo [34] cung cấp một khảo sát hoàn chỉnh với các chi tiết và tài liệu tham khảo.

Trong hai mươi năm kể từ khi những kết quả mang tính đột phát này xuất hiện, đã có một sự bùng nổ về các kết quả về sự không xấp xỉ được, nhận được thông qua việc khai thác các biến thể và cải tiến các kết quả PCP ban đầu, và dựa trên một loạt các cải tiến của giả thiết P $\neq$ NP. Để biết thêm chi tiết, xem các bài khảo sát, ví dụ như [36, 54]. Ngày nay, chúng ta biết rằng CLIQUE không thể được xấp xỉ đến một thừa số $n^{1-\epsilon}$ với bất kỳ hằng số $\epsilon > 0$ nào trừ khi P = NP [56]. Chúng ta cũng biết rằng thuật toán tham lam cho SET COVER đề cập ở trên không thể tốt hơn (không quan tâm đến các hạng tử bậc thấp (low-order terms) trong tỷ lệ xấp xỉ) trừ khi NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$ [19].

Các giả thuyết khác được sử dụng trong chứng minh các kết quả về sự khó khăn của xấp xỉ bao gồm NP $\not\subseteq$ DTIME$[n^{O(\log\log\log n)}]$, NP $\not\subseteq \cup_{k>0}$DTIME$[n^{\log^k n}]$, NP $\not\subseteq \cap_{\epsilon>0}$DTIME$[2^{n^\epsilon}]$, và NP $\not\subseteq$ BPP, với BPP là lớp các bài toán giải được bằng các thuật toán ngẫu nhiên trong thời gian đa thức. Tuy nhiên, hiện tại, giả thuyết phổ biến nhất là "giả thuyết các trò chơi độc đáo" ("Unique Games Conjecture" – UGC) của Subhash Khot [41]. Giả sử chúng ta có một số nguyên tố q cho trước, một số $\epsilon > 0$ nhỏ, và một danh sách các phương trình có dạng $x_j - x_k = c_h \pmod q$ với các biến x_i và các hằng số c_h. Giả thuyết trên nói rằng sẽ là NP-khó để phân biệt giữa trường hợp mà ít nhất một tỷ lệ $1-\epsilon$ trong số các phương trình có thể được thoả mãn đồng thời và trường hợp mà không có nhiều hơn một tỷ lệ ϵ trong số các phương trình có thể được thoả mãn đồng thời – một lỗ hổng rất lớn. Như với các kết quả PCP, giả thuyết này bắt nguồn từ một bài toán liên quan đến các hệ thống với nhiều người chứng minh, và ở trong bối cảnh này nó nhận được tên gọi như hiện tại.

Lý do mà giả thuyết đặc biệt này thu hút được nhiều sự chú ý là do nó suy ra rằng với nhiều bài toán quan trọng, các thuật toán xấp xỉ tốt nhất của chúng ta hiện tại không thể được cải tiến trừ khi P = NP. Ví dụ, không có thuật toán xấp xỉ nào chạy trong thời gian đa thức cho VERTEX COVER mà có thể đảm bảo tốt hơn thừa số của 2 đã được đảm bảo bởi rất nhiều thuật toán xấp xỉ đơn giản [9]. Tương tự, thuật toán Goemans-Williamson [24] cho MAX CUT, sử dụng quy hoạch nửa xác định (semidefinite programming) và làm tròn ngẫu nhiên (randomized rounding), và có đảm bảo trong trường hợp xấu nhất bằng $(2/\pi)/\min_{0<\theta\leq\pi}((1-\cos(\theta))/\theta) \sim .878$, không thể được cải tiến bởi bất kỳ thuật toán trong thời gian đa thức nào [42]. Tổng quát hơn, với bất kỳ

bài toán thoả mãn ràng buộc (Constraint Satisfaction Problem - CSP) nào mà mục tiêu là tìm một cách gán giá trị (assignment) cho các biến để thoả mãn số lượng lớn nhất các ràng buộc (constraint), có thể chỉ ra rằng một thuật toán cơ bản, dựa trên quy hoạch nửa xác định và làm tròn, đạt được tỷ lệ xấp xỉ trong trường hợp xấu nhất tốt nhất so với bất kỳ thuật toán trong thời gian đa thức nào khác, giả thiết P $\neq$ NP và UGC [47], và thậm chí mặc dù với rất nhiều bài toán như thế chúng ta ở thời điểm này chưa biết tỷ lệ xấp xỉ đó là bao nhiêu.

Giả thuyết UGC liệu có đúng hay không dĩ nhiên vẫn là một câu hỏi mở, và các nhà nghiên cứu có xu hướng hoài nghi hơn là giả thuyết P $\neq$ NP. Thêm vào đó, ảnh hưởng của nó dường như chỉ giới hạn trong các bài toán mà các thuật toán xấp xỉ với tỷ lệ hữu hạn trong trường hợp xấu nhất tồn tại, trong khi các giả thuyết khác nhắc đến ở trên đã dẫn đến nhiều chặn dưới không phải hằng số, ví dụ như chặn dưới $\ln n$ cho SET COVER. Điều này dẫn tới một tác dụng phụ thú vị khiến các thuật toán với tỷ lệ trong trường hợp xấu nhất không phải hằng số trở nên đáng tôn trọng hơn – nếu ta không thể làm tốt hơn $\Omega(\log n)$, thì có lẽ $O(\log^2 n)$ cũng không tệ lắm? Trên thực tế, một bài báo gần đây có một kết quả đột phá là bài toán LABEL COVER có một thuật toán xấp xỉ trong thời gian đa thức với tỷ lệ trong trường hợp xấu nhất $O(n^{1/3})$, đánh bại tỷ lệ tốt nhất trước đó $O(n^{1/2})$ [11].

Hãy để tôi kết luận bài báo này bằng một câu hỏi hiển nhiên. Tất cả những thứ này chắc chắn tạo nên một lý thuyết thú vị, nhưng điều này có ý nghĩa gì trong thực hành? Tôi tin tưởng rằng những năm qua đã dạy chúng ta phải thực sự nghiêm túc với những cảnh báo từ NP-đầy đủ. Nếu một bài toán tối ưu là NP-khó, hiếm khi chúng ta tìm được các thuật toán mà, thậm chí khi chỉ xét các trường hợp "thực tế", luôn luôn tìm được các lời giải tối ưu, và thực hiện điều đó trong thời gian đa thức trong thực hành (empirical polynomial time). Thậm chí thành tựu lớn nhất của lý thuyết tối ưu, chương trình CONCORDE để giải bài toán TSP một cách tối ưu [4], cần có thời gian chạy siêu đa thức (super-polynomial running time), ngay cả khi chỉ xét các trường hợp đơn giản bao gồm các điểm phân bố đều trên hình vuông đơn vị, trong đó thời gian chạy trung bình của nó có vẻ tăng theo hàm mũ của $\sqrt{n}$ [30].

Do đó, lời biện minh cũ cho việc chuyển sang các thuật toán xấp xỉ vẫn còn hiệu nghiệm. Việc nó có thể được giải thích như thế nào bằng các kết quả về độ khó khăn của xấp xỉ có vẻ không rõ ràng lắm. Rất nhiều thuật toán xấp xỉ, ví dụ như thuật toán tham lam cho SET COVER, có vẻ như tiếp cận tối ưu gần hơn nhiều so với các chặn trong trường hợp xấu nhất của họ thể hiện, và chỉ bởi vì một bài toán khó có thể xấp xỉ trong trường hợp xấu nhất về mặt lý thuyết không có nghĩa là chúng ta không thể nghĩ ra các thuật toán heuristics để tìm các lời giải tương đối tốt trong thực hành. Và thực sự là, khi việc tối ưu chính xác không còn phù hợp, chúng ta còn lựa chọn nào khác hơn là tìm kiếm chúng?

Tài liệu

[1] http://www.nsa.gov/public_info/_files/nash_letters/nash_letters1.pdf

[2] http://www.gwern.net/docs/1955-nash

[3] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Ann. Math.*, 160:781–793, 2004. Journal version of a 2002 preprint.

[4] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, editors. *The Traveling Salesman Problem*. Princeton University Press, Princeton, NJ, 2006.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. In *Proc. 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 14–23, Los Alamitos, CA, 1992. IEEE Computer Society. Journal version, see [6].

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation algorithms. *J. ACM*, 45(3):501–555, 1998.

[7] S. Arora and S. Safra. Probabilistically checkable proofs; a new characterization of NP. In *Proc. 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 2–13, Los Alamitos, CA, 1992. IEEE Computer Society. Phiên bản tạp chí, xem [8].

[8] S. Arora and S. Safra. Probabilistically checkable proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[9] P. Austrin, S. Khot, and M. Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011.

[10] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd Ann. ACM Symp. on Theory of Computing*, pages 21–31, New York, 1991. Association for Computing Machinery.

[11] M. Charikar, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61:190–206, 2011.

[12] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In *Symposium on New Directions and Recent Results in Algorithms and Complexity*, J.F. Traub, (ed.), page 441. Academic Press, NY, 1976.

[13] A. Cobham. The intrinsic computational difficulty of functions. In Y. BarHillel, editor, *Proc. 1964 International Congress for Logic Methodology and Philosophy of Science*, pages 24–30, Amsterdam, 1964. North Holland.

[14] S. Cook. The complexity of theorem proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.

[15] S. A. Cook. Deterministic CFL's are accepted simultaneously in polynomial time and log squared space. In *Proc. 11th Ann. ACM Symp. on Theory of Computing*, pages 338–345, New York, 1979. Association for Computing Machinery.

[16] D. P. Dobkin, R. J. Lipton, and S. P. Reiss. Linear programming is log space hard for P. *Inf. Proc. Lett.*, 8(2):96–97, 1979.

[17] J. Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. Nat. Bur. Standards Sect. B*, 69:67–72, 1965.

[18] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math*, 17:449–467, 1965.

[19] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998. (Preliminary version in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1996, 314–318.).

[20] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. 32nd Ann. IEEE Symp. on Foundations of Computer Science*, pages 2–12, Los Alamitos, CA, 1991. IEEE Computer Society.

[21] M. R. Garey, R. L. Graham, and J. D. Ullman. Worst-case analysis of memory allocation algorithms. In *Proc. 4th Ann. ACM Symp. on Theory of Computing*, pages 143–150, New York, 1972. Association for Computing Machinery.

[22] M. R. Garey and D. S. Johnson. Strong NP-completeness results: Motivation, examples, an implications. *J. ACM*, 25(3):499–508, 1978.

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, New York, 1979.

[24] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995. (Preliminary version in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1994, 422–431.).

[25] M. Goldberg, V. Lifschitz, and B. Trakhtenbrot. *A Colloquium on Large Scale Finite Mathematics in the U.S.S.R*. Delphi Associates, Falls Church, VA, 1984. Đây là bản tóm tắt của một cuộc thảo luận mà tôi tham dự và là một trong số mà tôi có một bản thảo ban đầu. Rất nhiều trang web liệt kê nó như là một quyển sách với một số ISBN và có cùng số trang với bản thảo ban đầu của tôi, và Google đưa ra một bức ảnh của có thể là một bản in bìa cứng, nhưng dường như không có ai bán nó.

[26] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo, editors. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, New York, 1995.

[27] J. Hartmanis. The structural complexity column: Gödel, von Neumann and the P=?NP problem. *Bull. European Assoc. for Theoretical Comput. Sci.*, 38:101–107, 1989.

[28] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandum generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[29] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, 1997.

[30] H. H. Hoos and T. Stützle, 2009. Private Communication.

[31] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975.

[32] D. S. Johnson. *Near-Optimal Bin Packing Algorithms*. PhD thesis, Massachusetts Institute of Technology, 1973.

[33] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comp. Syst. Sci.*, 9:256–278, 1974.

[34] D. S. Johnson. The NP-completeness column: An ongoing guide – the tale of the second prover. *J. Algorithms*, 13:502–524, 1992.

[35] D. S. Johnson. The NP-completeness column. *ACM Trans. Algorithms*, 1(1):160–176, 2005.

[36] D. S. Johnson. The NP-completeness column: The many limits on approximation. *ACM Trans. Algorithms*, 2(3):473–489, 2006.

[37] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, New York, 1972. Plenum Press.

[38] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.

[39] W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. *J. ACM*, 55(2):Article A11, 2008.

[40] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[41] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, New York, 2002. Association for Computing Machinery.

[42] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

[43] D. E. Knuth. A terminological proposal. *SIGACT News*, 6(1):12–18, 1974.

[44] L. A. Levin. Universal sequential search problems. *Problemy Peredachi Informatskii*, 9(3):115–116, 1973.

[45] L. A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

[46] R. E. Miller and J. W. Thatcher, editors. *Complexity of Computer Computations*. Plenum Press, New York, 1972.

[47] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–154, New York, 2008. Association for Computing Machinery.

[48] S. Sahni. Approximate algorithms for the 0/1 knapsack problem. *J. ACM*, 22(1):115–124, 1975.

[49] S. Sahni and T. Gonzalez. P-complete problems and approximate solutions. In *Proc. 15th Ann. IEEE Symp. on Foundations of Computer Science*, pages 28–32, Los Alamitos, CA, 1974. IEEE Computer Society. Một bài báo đăng trên tạp chí mở rộng các kết quả không xấp xỉ được (inapproximability results) trong bài báo này là [50].

[50] S. Sahni and T. Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, 1976.

[51] D. Shasha and C. Lazere. *Out of their Minds*. Copernicus, New York, 1995.

[52] B. A. Trakhtenbrot. A survey of Russian approaches to *perebor* (bruteforce search) algorithms. *Ann. History of Computing*, 6:384–400, 1984.

[53] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.

[54] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, 2011.

[55] B. Yamnitsky and L. A. Levin. An old linear programming algorithm runs in polynomial time. In *Proc. 23rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 327–328, Los Alamitos, CA, 1982. IEEE Computer Society.

[56] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 681–690, New York, 2006. Association for Computing Machinery.