

Tìm hiểu một số phương pháp giải quyết bài toán NP - khó

Nguyễn Ngọc Bảo An*

* Lớp K68N, Trường Đại học Công nghệ - ĐHQGHN

Received: 28/11/2023; Accepted: 6/12/2023; Published: 05/01/2024

Abstract: The NP-hard problem is a very common problem in life such as the tourist problem, the backpack packing problem, the scheduling problem, etc. However, up to now all the research At home and abroad, we still have not found an exact solution in polynomial time to the NP-hard problem. Therefore, the author approaches to learn some methods for solving NP-hard problems that are appropriate and highly applicable.

Keywords: Difficult NP problem, solved

1. Đặt vấn đề

Bài toán NP-khó là bài toán thường gặp rất nhiều trong cuộc sống như bài toán người đi du lịch, bài toán xếp ba lô, bài toán xếp lịch,... Tuy nhiên, cho đến hiện nay tất cả các nghiên cứu trong và ngoài nước vẫn chưa tìm được lời giải chính xác trong thời gian đa thức cho bài toán NP-khó. Vì vậy, tác giả tiếp cận tìm hiểu một số phương pháp giải quyết bài toán NP-khó là phù hợp và có tính ứng dụng cao.

2. Nội dung nghiên cứu

2.1. Bài toán NP-khó

2.1.1. Lớp bài toán P, NP và co-NP

Dưới đây là phân loại các lớp của bài toán:

Định nghĩa 1. P là lớp bài toán quyết định có thể được giải quyết trong thời gian đa thức.

Hay nói cách khác, P là lớp các bài toán có thể được giải một cách nhanh chóng.

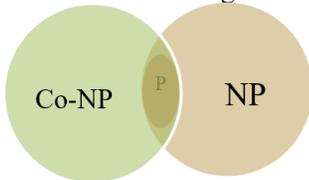
Định nghĩa 2. NP là lớp bài toán quyết định mà để xác nhận câu trả lời là "yes" của nó, có thể đưa ra bằng chứng ngắn gọn dễ kiểm tra.

Hay có thể nói NP là lớp bài toán mà có thể kiểm tra câu trả lời "yes" một cách nhanh chóng trong thời gian đa thức nếu đã có được lời giải.

Hiện nhiên ta có $P \subset NP$, tuy nhiên xác định xem $NP \subset P$ hay không hiện vẫn chưa có lời giải.

Định nghĩa 3 co-NP là lớp bài toán mà để xác nhận câu trả lời "no" thì có thể đưa ra bằng chứng ngắn gọn dễ kiểm tra.

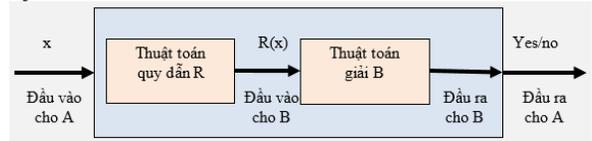
Như vậy có thể thấy co-NP là lớp bài toán hoàn toàn ngược với lớp NP. Có thể miêu tả mối quan hệ giữa ba lớp bài toán trên như trong hình dưới đây:



Hình 2.1: Các lớp bài toán P, NP và co-NP

2.2. Khái niệm quy dẫn

Định nghĩa 4 Giả sử A và B là hai bài toán quyết định. Ta nói bài toán A có thể quy dẫn sau thời gian đa thức về bài toán B nếu tồn tại thuật toán thời gian đa thức R cho phép biến đổi bộ dữ liệu vào x của A thành bộ dữ liệu vào R(x) của B sao cho x là bộ dữ liệu "yes" của A khi và chỉ khi R(x) là bộ dữ liệu "yes" của B.



Hình 2.2: Sơ đồ quá trình quy dẫn

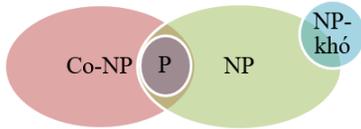
Ký hiệu $A < B$ được dùng để chỉ bài toán A có thể quy dẫn về bài toán B. Phép quy dẫn thường dùng để so sánh độ khó của hai bài toán. Nếu A quy dẫn được về B thì A không khó hơn B. Nếu A là khó (theo nghĩa chưa tìm được thuật toán thời gian tính đa thức để giải A) thì B cũng là khó, còn nếu B là dễ (nghĩa là đã có thuật toán thời gian tính đa thức giải B) thì A cũng là dễ.

2.3. Lớp bài toán NP-đầy đủ và NP-khó

Định nghĩa 5 Một bài toán quyết định A được gọi là NP-đầy đủ nếu như A là bài toán trong NP và mọi bài toán trong NP đều có thể quy dẫn về A.

Định nghĩa 6 Một bài toán A được gọi là NP-khó nếu như sự tồn tại thuật toán đa thức để giải nó kéo theo sự tồn tại thuật toán đa thức để giải mọi bài toán trong NP.

Nói cách khác, nếu có thể giải một bài toán NP-khó nào đó một cách nhanh chóng, thì cũng có thể nhanh chóng giải quyết bất kỳ một bài toán nào khác. Bài toán NP-khó ít nhất là khó bằng bất cứ một bài toán nào trong NP. NP-đầy đủ là những bài toán khó nhất trong NP. Hình dưới đây biểu diễn cách phân loại tạm thời các bài toán.



Hình 2.3: Phân lớp tạm thời các bài toán

2.4. Một số phương pháp giải quyết bài toán NP-khó

Mục đích của bài toán tối ưu tổ hợp là tìm lời giải tốt nhất trong các lời giải có thể và không gian tìm kiếm lời giải của bài toán là rời rạc. Nhiều bài toán tổ hợp có độ phức tạp tính toán lớn và được phân loại vào lớp bài toán NP-khó (không tìm được lời giải tối ưu trong thời gian đa thức). Việc tìm ra lời giải tối ưu cho loại bài toán này trong các hệ thống song song lớn nhất cũng không thể thực hiện trong giới hạn thời gian cho phép, vì vậy các giải thuật heuristic, meta-heuristic giải các bài toán tổ hợp theo hướng xấp xỉ đã được phát triển để tìm ra lời giải gần tối ưu trong thời gian chấp nhận được.

Meta-heuristic là một cách gọi chung cho các giải thuật heuristic trong việc giải quyết các bài toán tổ hợp khó. Meta-heuristic bao gồm những chiến lược khác nhau trong việc khám phá không gian tìm kiếm bằng cách sử dụng những phương thức khác nhau và phải đạt được sự cân bằng giữa tính đa dạng và chuyên sâu của không gian tìm kiếm. Việc cài đặt thành công một giải thuật heuristic cho một bài toán tổ hợp đòi hỏi phải cân bằng giữa sự khai thác kinh nghiệm thu thập được trong quá trình tìm kiếm để xác định được những vùng với lời giải có chất lượng cao gần tối ưu. Các giải thuật heuristic và meta-heuristic nổi tiếng được biết đến như giải thuật mô phỏng luyện kim (SA- Simulated Annealing), giải thuật di truyền (GA - Genetic Algorithm), giải thuật tối ưu hóa đàn kiến (ACO – Ant Colony Optimization) ...

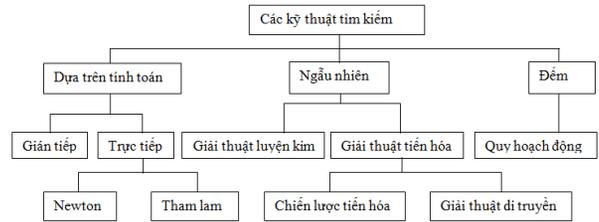
❖ **Giải thuật tối ưu hóa đàn kiến (ACO)** được đề xuất bởi Marco Dorigo năm 1992 là meta-heuristic dùng chiến lược của đàn kiến trong thế giới thực để giải bài toán tối ưu.

❖ **Giải thuật di truyền (GA)** do John Holland phát minh và được ông phát triển cùng với các đồng nghiệp và sinh viên. Giải thuật xuất phát từ phương thức xác suất dựa trên ý tưởng từ cơ chế di truyền trong sinh học và tiến trình tiến hóa trong cộng đồng cá thể của một loài.

❖ **Giải thuật mô phỏng luyện kim (SA)** là phương pháp xác suất được đề xuất bởi Kirkpatrick, Gelett và Vecchi (1983) và Cerny (1985) lại sử dụng tính chất thu được trạng thái năng lượng nhỏ nhất

nhờ nóng chảy và ngưng tụ của kim loại trong tự nhiên để tìm ra giải pháp tốt nhất cho các loại bài toán tối ưu này.

Hình sau đây cho ta thấy mối liên hệ giữa các kỹ thuật tìm kiếm



Hình 2.4: Các kỹ thuật tìm kiếm

2.4.1. Giải thuật heuristic

Heuristic là phương pháp giải quyết vấn đề bằng cách đánh giá kinh nghiệm và tìm giải pháp qua thử nghiệm và hạn chế khuyết điểm. Từ heuristic sử dụng trong thuật toán dùng để tìm giải pháp trong số những cái có thể, nhưng không đảm bảo rằng cái tốt nhất sẽ được tìm thấy, do đó họ có thể giả định đó là thuật toán xấp xỉ và không chính xác. Các thuật toán này thường tìm ra một giải pháp gần như tốt nhất và thường chúng tìm thấy nhanh chóng và dễ dàng. Tinh thoảng các thuật toán này có thể chính xác và thực sự tìm và giải pháp tốt nhất, nhưng thuật toán vẫn được gọi là heuristic, cho đến khi giải pháp tốt nhất này được chứng minh là đưa ra được lời giải tối ưu.

Có nhiều phương pháp để xây dựng một giải thuật heuristic, trong đó người ta thường dựa vào một số nguyên lý cơ sở sau:

❖ Nguyên lý vét cạn thông minh:

Trong bài toán tìm kiếm nào đó, khi không gian tìm kiếm lớn, ta thường tìm cách giới hạn lại không gian tìm kiếm hoặc thực hiện một kiểu dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.

❖ Nguyên lý tham lam:

Lấy tiêu chuẩn tối ưu (trên phạm vi toàn cục) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ từng bước trong quá trình tìm kiếm lời giải.

❖ Nguyên lý thứ tự:

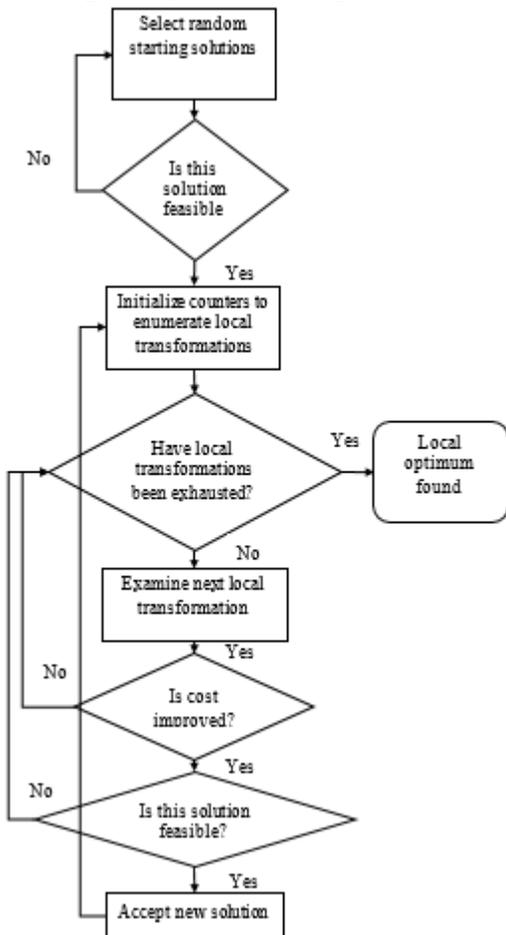
Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt nhất.

❖ Hàm heuristic:

Trong việc xây dựng các giải thuật heuristic, người ta thường dùng các hàm Heuristic. Đó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào

trạng thái của bài toán tại mỗi bước giải. Nhờ giá trị này, ta có thể chọn được các di chuyển trong đối hợp lý trong từng bước giải của giải thuật.

Mô hình giải thuật heuristic tổng quát như sau:



Hình 2.5. Sơ đồ khối giải thuật heuristic tổng quát

2.4.2. Giải thuật tìm kiếm địa phương

Tìm kiếm địa phương là một giải thuật thuộc lớp meta heuristic điển hình và đã được áp dụng với nhiều bài toán tối ưu tổ hợp. Ý tưởng của giải thuật này là xuất phát từ một lời giải chấp nhận được ban đầu (lời giải này thỏa mãn hết các ràng buộc của bài toán, nhưng có thể chưa phải là lời giải tối ưu). Mỗi bước lặp sẽ di chuyển đến một lời giải láng giềng của lời giải hiện tại sao cho lời giải láng giềng tối ưu hơn lời giải hiện tại. Giải thuật kết thúc nếu như không tìm được lời giải láng giềng nào tối ưu hơn lời giải hiện tại. Điểm mấu chốt của giải thuật tìm kiếm địa phương là xây dựng tập các lời giải láng giềng. Kích thước và sự đa dạng của tập này phải hợp lý để đảm bảo có thể tìm ra được lời giải tối ưu hơn trong thời gian cho phép.

Giải thuật Local Search

```

1.  $S_0 \leftarrow$  lời giải khởi tạo
2. Repeat
3.   stop  $\leftarrow$  true
4.   For  $S_i \in Neighbor\_Solution(S_0)$  do
5.     If  $S_i$  tối ưu hơn  $S_0$  then
6.        $S_0 \leftarrow S_i$ 
7.       stop  $\leftarrow$  false
8.       break
9.     End If
10.  End for
11. Until stop
12. Return  $S_0$ 

```

2.4.3. Giải thuật tabu search

Giải thuật tabu search được Fred W. Glover giới thiệu vào năm 1986 để khắc phục những nhược điểm của tìm kiếm địa phương.

Giải thuật tìm kiếm địa phương đưa ra một lời giải hợp lệ ban đầu cho bài toán và xét các lời giải láng giềng với hy vọng tìm kiếm một lời giải tốt hơn lời giải hiện tại. Tìm kiếm địa phương có xu hướng tối ưu cục bộ hoặc lặp rất nhiều lần khi các lời giải láng giềng là như nhau. Tabu search nâng cao hiệu suất tìm kiếm địa phương bằng cách nói lỏng quy tắc cơ bản của nó. Đầu tiên, tabu search cho phép di chuyển đến một lời giải tốt hơn nếu như không thể tìm được một lời giải láng giềng tốt hơn lời giải hiện tại (trường hợp đạt đến tối ưu cục bộ). Thứ hai, nếu một lời giải tiềm năng đã được truy cập trước đó trong một khoảng thời gian ngắn nhất định thì lời giải này được đưa vào danh sách cấm, không được phép di chuyển từ lời giải hiện tại đến một lời giải nằm trong danh sách cấm.

3. Kết luận

Bài báo đã giới thiệu và tìm hiểu bài toán NP khó; Tìm hiểu một số cách tiếp cận giải bài toán NP khó; Đưa ra một số phương pháp giải quyết các bài toán NP khó (theo giải thuật heuristic, tìm kiếm địa phương và tabu search)

Từ đó có thể áp dụng một số phương pháp tiếp cận giải bài toán NP khó để giải bài toán người đi du lịch, bài toán xếp ba lô, bài toán xếp lịch, cũng như nhiều bài toán thực tế khác.

Tài liệu tham khảo

[1] L. Kou, G. Markowsky, and L. Berman. (1981), A fast algorithm for Steiner trees, Acta Informatica, vol. 15, no. 2, pp. 141–145.
 [2] D. S. Johnson. (1973), Approximation algorithms for combinatorial problems, in Proc. ACM STOC, pp. 38–49.
 [3] M. R. Garey, and D. S. Johnson. (1979), Computers and Intractability: A Guide to the Theory of NPCompleteness, W. H. Freeman, New York.