

AI

CHỨC NĂNG HEURISTIC

Nguyễn Ngọc Thảo - Nguyễn Hải Minh
{nnthao, nhminh}@fit.hcmus.edu.vn

Bài toán 8 câu đố

Trạng thái bắt đầu

7	2	4
5		6
8	3	1

Trạng thái mục tiêu

	1	2
3	4	5
6	7	8

Một ví dụ điển hình của câu đố 8. Lời giải dài 26 bước.

- Chi phí giải pháp trung bình: khoảng 22 bước, hệ số phân nhánh ~ 3 .
- 15 câu đố: $16! / 2$ —trên 10 nghìn tỷ tiểu bang • 24 câu

đố: khoảng 1025 tiểu bang

Các phương pháp phỏng đoán được chấp nhận cho 8 câu đố

- $h_1(n)$ = số ô được đánh số sai vị trí (Khoảng cách Hamming).



- $h_2(n)$ = tổng khoảng cách (Manhattan) của mỗi ô được đánh số tới vị trí mục tiêu của nó.
- Cả hai điều này đều không đánh giá quá cao chi phí thực sự của giải pháp-26.

Câu đố 03: Phương pháp phỏng đoán có thể chấp nhận được

- Đối với trò chơi 8 câu đố, phương pháp suy nghiệm nào sau đây được chấp nhận?

- $h1() =$ tổng số ô bị đặt sai vị trí •

$h2() =$ tổng khoảng cách Manhattan

- $h3() = 0$ •

$h4 = (1)$

$h5() = h()$

- $h6() = \min(2, h$

$())$ • $h7() = \max(2, h())$

Ảnh hưởng của heuristic đến hiệu suất

- Một cách để mô tả chất lượng của một phương pháp phỏng đoán là **hệ số phân nhánh hiệu quả**.
- Đặt tổng số nút được tạo bởi A^* cho một bài toán cụ thể và độ sâu lời giải là .
- là hệ số phân nhánh mà một cây có độ sâu đồng nhất sẽ phải chứa $+1$ nút.
- Như vậy,
$$+ = + \quad * + (\quad) + \cdot \cdot \cdot + (\quad)$$
- Ví dụ: A^* tìm giải pháp ở độ sâu 5 bằng 52 nút $* = 1,92$

Ảnh hưởng của heuristic đến hiệu suất

- * có thể khác nhau giữa các trường hợp vấn đề, nhưng không đòi hỏi trong tất cả các trường hợp không cần thiết trong một miền cụ thể.
- Các thử nghiệm trên một tập hợp nhỏ các bài toán có thể mang lại kết quả hướng dẫn tốt về tính hữu dụng tổng thể của heuristic.
- Một heuristic được thiết kế tốt sẽ có giá trị gần với 1, cho phép giải quyết các vấn đề lớn với chi phí hợp lý.

Chi phí tìm kiếm so với hệ số phân nhánh

d	Search Cost (nodes generated)			Effective Branching Factor		
	BFS	$A^*(h_1)$	$A^*(h_2)$	BFS	$A^*(h_1)$	$A^*(h_2)$
6	128	24	19	2.01	1.42	1.34
8	368	48	31	1.91	1.40	1.30
10	1033	116	48	1.85	1.43	1.27
12	2672	279	84	1.80	1.45	1.28
14	6783	678	174	1.77	1.47	1.31
16	17270	1683	364	1.74	1.48	1.32
18	41558	4102	751	1.72	1.49	1.34
20	91493	9905	1318	1.69	1.50	1.34
22	175921	22955	2548	1.66	1.50	1.34
24	290082	53039	5733	1.62	1.50	1.36
26	395355	110372	10080	1.58	1.50	1.35
28	463234	202565	22055	1.53	1.49	1.36

So sánh chi phí tìm kiếm và các hệ số phân nhánh hiệu quả cho các bài toán 8 câu đố sử dụng BFS, A^* với các ô đặt sai vị trí (h_1) và A^* với khoảng cách Manhattan (h_2). Dữ liệu được tính trung bình trên 100 câu đố cho mỗi lời giải có độ dài từ 6 đến 28.

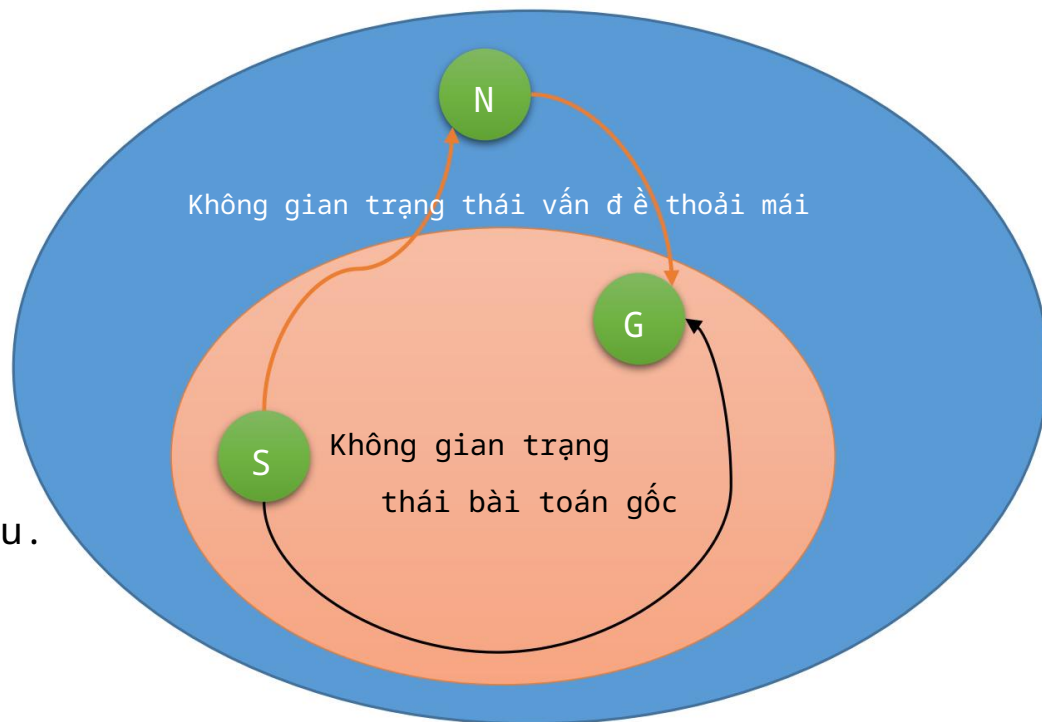
Sự thống trị heuristic

- Xét hai phương pháp phỏng đoán h_1 và h_2 .
- h_2 trội hơn h_1 nếu $h_2(n) \geq h_1(n)$, với mọi n .
- A^* sử dụng h_2 sẽ không bao giờ mở rộng nhiều nút hơn A^* sử dụng h_1 .
 - Giả sử h_1 và h_2 đều nhất quán và $h_2(n) \geq h_1(n)$.
 - Đối với A^* , mọi nút có $f(n) < f^*$ (tức là $h(n) < f^* - g(n)$) được mở rộng, chắc chắn sẽ khi h nhất quán.
 - Do đó, mọi nút chắc chắn được mở rộng bởi A^* với h_2 cũng chắc chắn được mở rộng bằng h_1 và h_1 có thể khiến các nút khác được mở rộng.
- Một heuristic có giá trị cao hơn sẽ hữu ích hơn, miễn là nó nhất quán và thời gian tính toán không quá dài.

Heuristic từ các vấn đề thoả mái

- Bài toán có ít hạn chế hơn đối với các hành động được gọi là bài toán **vấn đề thoả mái**.

Đồ thị không gian trạng thái của bài toán thoả mái là siêu đồ thị của không gian trạng thái ban đầu.



- Chi phí của một giải pháp tối ưu cho một bài toán thoả mái là một heuristic nhất quán cho vấn đề ban đầu.

Suy nghiệm từ câu đố 8 câu thoả mái

Một ô có thể di chuyển từ hình vuông A sang hình vuông B nếu A nằm cạnh B theo chiều ngang hoặc chiều dọc và B trống

- Các vấn đề dễ dàng được tạo ra bằng cách loại bỏ một hoặc cả hai điều kiện
 - Một ô có thể di chuyển từ ô A sang ô B nếu ô A liền kề với ô B.
 - Một ô có thể di chuyển từ ô A sang ô B nếu ô B trống.
 - Một ô có thể di chuyển từ ô A sang ô B.

Gạch đặt sai
vị trí

khoảng cách
Manhattan

Vấn đề thư giãn

- Các vấn đề thoải mái được tạo ra bằng cách loại bỏ các hạn chế phải về cơ bản có thể được **giải quyết mà không cần tìm kiếm**.
- Điều này cho phép bài toán ban đầu được **phân tách thành một số bài toán con độc lập**.
 - Ví dụ: giải bài toán 8 câu đố sẽ có 8 câu đố độc lập các vấn đề con, mỗi vấn đề một ô.
- Nếu bài toán thoải mái **khó giải thì** việc thu được các giá trị của **heuristic tương ứng sẽ rất tốn kém**.

Heuristic tổng hợp

- Xem xét một tập hợp các phương pháp phỏng đoán có thể chấp nhận được cho một vấn đề, $\{h_1, h_2, \dots, h_n\}$ — không ai rõ ràng là tốt hơn những người khác.
- **Heuristic tổng hợp** được định nghĩa như sau.

$$H = \{h_1, h_2, \dots, h_n\}$$
- Heuristic này **nhất quán và chiếm ưu thế** trong tất cả các thành phần heuristics, nhưng **phải mất nhiều thời gian hơn để tính toán**.
 - Thay thế: chọn ngẫu nhiên một trong các phương pháp phỏng đoán ở mỗi lần đánh giá hoặc sử dụng thuật toán ML để dự đoán phương pháp phỏng đoán nào sẽ tốt nhất.
 - Điều này có thể dẫn đến một phương pháp phỏng đoán không nhất quán nhưng thực tế lại dẫn đến giải quyết vấn đề nhanh hơn.

Heuristic từ các bài toán con

- Các phương pháp phỏng đoán có thể chấp nhận được cũng có thể được rút ra từ lời giải chi phí của một bài toán con của một bài toán nhất định.
- Chi phí này là **giới hạn dưới** của chi phí của toàn bộ vấn đề.
- Chính xác hơn** khoảng cách Manhattan trong một số trường hợp.

Trạng thái bắt đầu

*	2	4
*		*
*	3	1

Trạng thái mục tiêu

	1	2
3	4	*
*	*	*

Nhiệm vụ là đưa các ô 1, 2, 3, 4 và ô trống vào đúng vị trí của chúng mà không phải lo lắng về điều gì sẽ xảy ra với các ô khác.

Heuristic từ các bài toán con

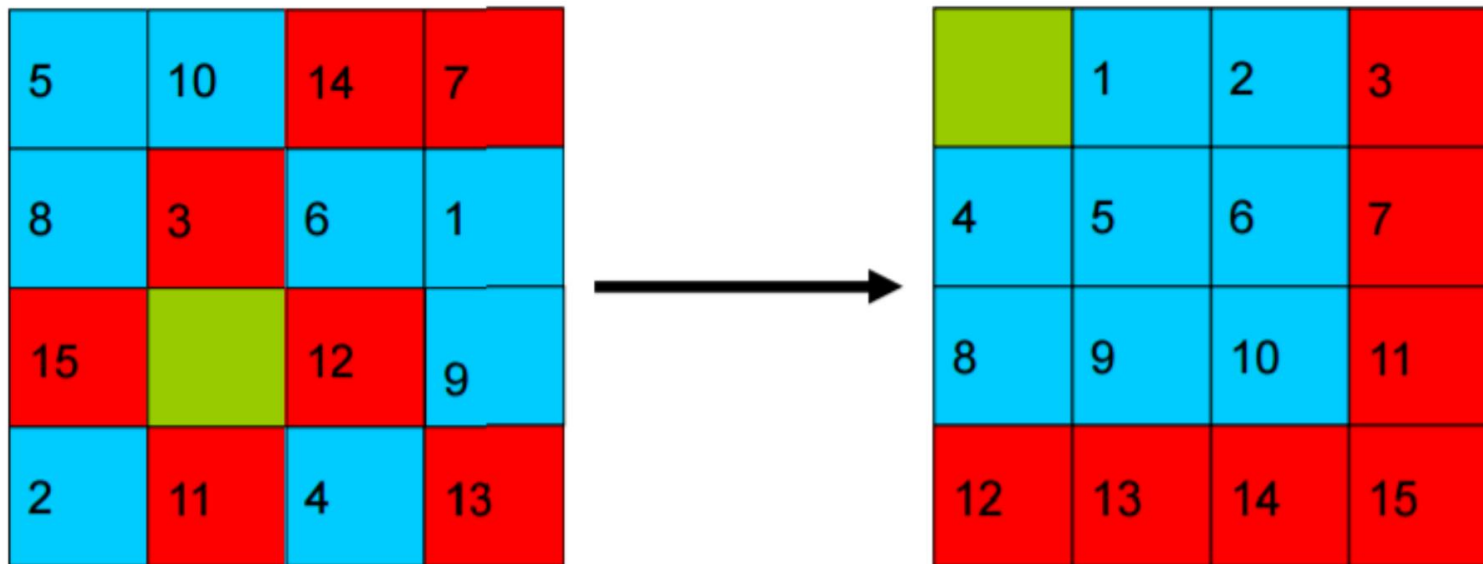
- Cơ sở dữ liệu mẫu (PDB) lưu trữ chi phí giải pháp chính xác cho mọi trường hợp bài toán con có thể xảy ra.
 - Ví dụ: cấu hình 8 câu đố trước đây có $9 \times 8 \times 7 \times 6 \times 5 = 15.120$ mẫu—mọi cách sắp xếp có thể có của bốn ô và ô trống.
- Cơ sở dữ liệu được xây dựng bằng cách tìm kiếm ngược từ mục tiêu và ghi lại chi phí của từng mẫu mới gặp phải.
 - Chi phí tìm kiếm được khấu hao trong các lần tiếp theo và
Vì vậy, sẽ hợp lý nếu chúng ta giải quyết được nhiều vấn đề.
- Chúng tôi xác định các giá trị heuristic cho các trạng thái bằng cách tham chiếu cấu hình vấn đề con có liên quan trong cơ sở dữ liệu.

Xuất bản trên cơ sở dữ liệu mẫu

- Culberson, Joseph C., và Jonathan Schaeffer " Cơ sở dữ liệu mẫu" Trí tuệ tính toán 14.3 (1998): 318-334.
- Korf, Richard E., và Ariel Felner: "Cơ sở dữ liệu mẫu rời rạc heuristic." Trí tuệ nhân tạo 134.1-2 (2002): 9-22.
- Felner, Ariel, Richard E. Korf, và Sarit Hanan " Heuristics cơ sở dữ liệu mẫu bổ sung" Tạp chí Nghiên cứu Trí tuệ Nhân tạo 22 (2004): 279-318.

cơ sở dữ liệu mẫu truyền thống

- Các phương pháp phỏng đoán có thể được kết hợp bằng cách **lấy giá trị lớn nhất**.
- Tuy nhiên, với mỗi DB bổ sung, **lợi nhuận giảm dần** và chi phí tính toán và bộ **nhớ tăng lên**.



Cần 31 bước để giải được gạch đỏ. Cần 22 bước để giải được gạch xanh

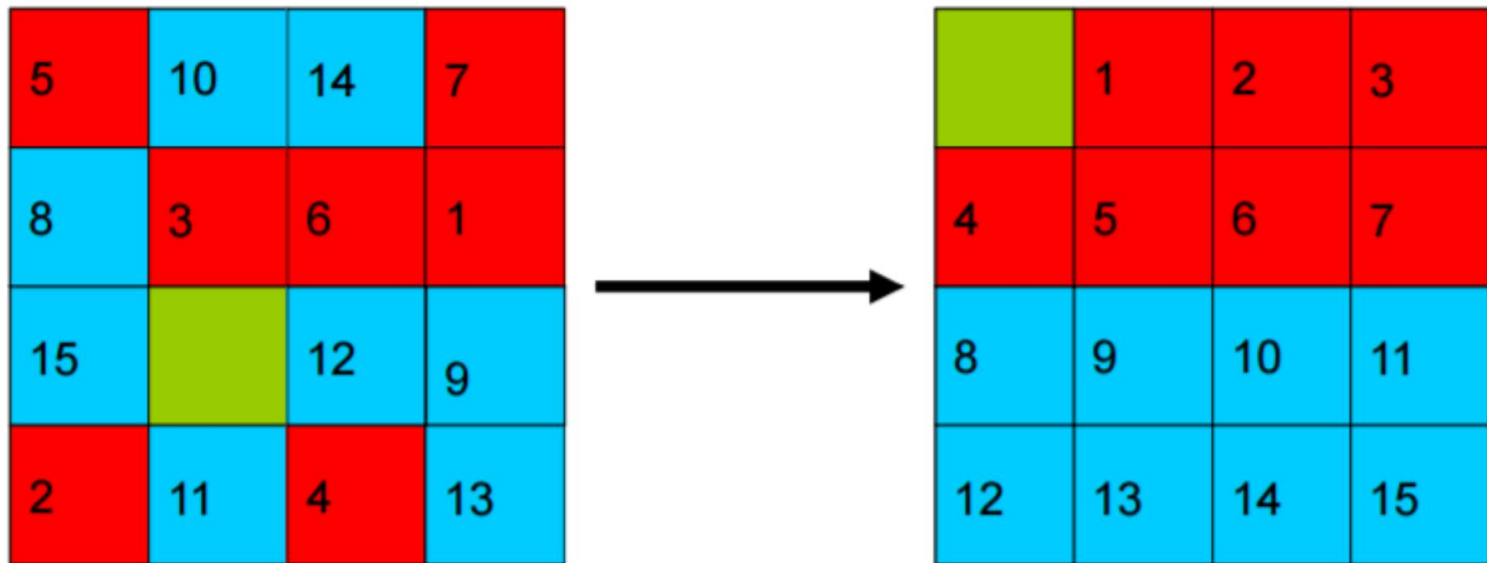
Heuristic tổng thể là giá trị tối đa - 31 nước đi

Cơ sở dữ liệu mẫu rời rạc

- Giả sử rằng không có ô nào thuộc nhiều hơn một mẫu.
- Cơ sở dữ liệu mẫu rời rạc chỉ tính các bước di chuyển của các ô mẫu trong khi bỏ qua các bước di chuyển không theo mẫu.
 - Nghĩa là, chúng ta ghi lại không phải tổng chi phí để giải bài toán con 1-2-3-4 mà là chỉ là số lần di chuyển liên quan đến 1-2-3-4.
- Có thể tổng hợp các chi phí, vốn vẫn là giới hạn dưới của chi phí để giải quyết toàn bộ vấn đề.

Cơ sở dữ liệu mẫu rời rạc

- Trong quá trình tìm kiếm, chúng tôi tra cứu các giá trị heuristic cho từng bộ và thêm các giá trị mà không đánh giá quá cao.



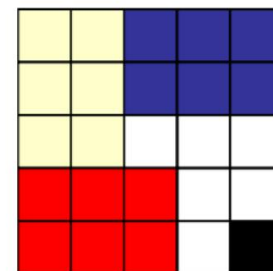
Cần 20 bước để giải được ô màu đỏ. Cần 25 bước để giải được ô màu xanh.

Heuristic tổng thể là $20 + 25 = 45$ nước đi

Heuristic từ các bài toán con

- Chúng ta có thể tổ chức nhiều cơ sở dữ liệu mẫu, mỗi cơ sở dữ liệu có kích thước rất lớn.
- Ví dụ: có hai cơ sở dữ liệu mẫu rời rạc cho 15 câu đố, một chứa 58 triệu mục và cái còn lại có 519 mục.

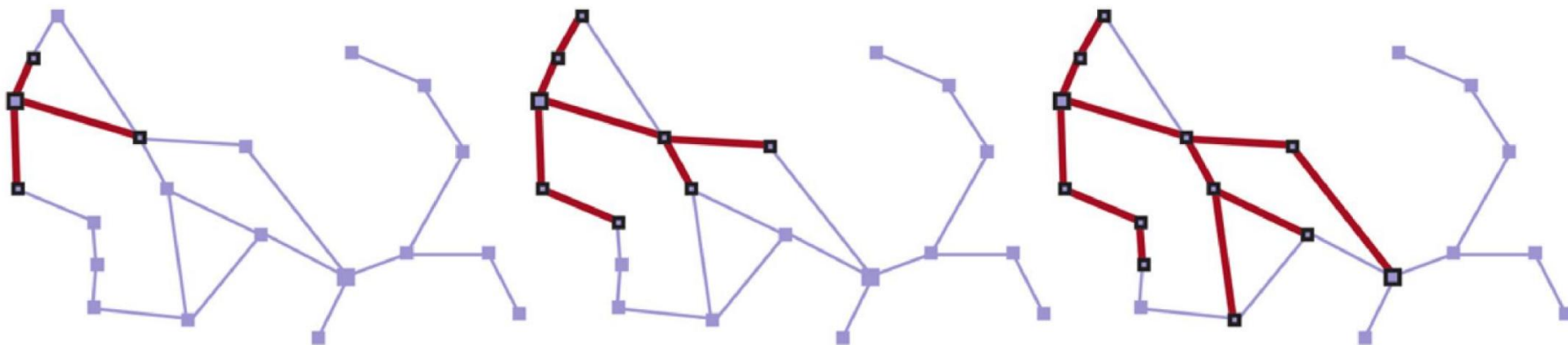
15- câu đố	<ul style="list-style-type: none"> • Tăng tốc 2000 so với khoảng cách Manhattan • IDA* với hai DB giải được 15 câu đố một cách tối ưu trong 30 mili giây
24- câu đố	<ul style="list-style-type: none"> • Tăng tốc 12 triệu so với Manhattan, không có PDB: 65.000 năm • IDA* có thể giải quyết các trường hợp ngẫu nhiên trong 2 ngày. • Yêu cầu 4 DB, mỗi DB có 128 triệu mục



Học cách tìm kiếm tốt hơn

- Đại lý có thể học cách tìm kiếm tốt hơn không ?
- Mỗi trạng thái trong không gian trạng thái siêu cấp độ nắm bắt trạng thái bên trong (tính toán) của chương trình đang tìm kiếm trong không gian trạng thái cấp đối tượng.
 - Trạng thái bên trong của thuật toán tìm kiếm là cây tìm kiếm hiện tại.
- Hành động là một bước tính toán làm thay đổi trạng thái bên trong.
 - Ví dụ: mở rộng một nút lá và thêm các nút kế tiếp của nó vào cây

Học cách tìm kiếm tốt hơn



Hình này mô tả một đường dẫn trong không gian trạng thái siêu cấp độ trong đó mỗi trạng thái trên đường dẫn (hình con) là một cây tìm kiếm cấp đối tượng.

- Việc mở rộng Fagoras không hữu ích thậm chí có thể xảy ra những vấn đề khó khăn hơn n bao gồm nhiều sai lầm như vậy.
- Thuật toán học siêu cấp thu được lợi ích từ những trải nghiệm này để tránh khám phá các cây con không có triển vọng học tăng cường.

Học heuristic từ kinh nghiệm

- Kinh nghiệm có nghĩa là giải quyết được rất nhiều trường hợp của một vấn đề.
 - Trứng, giải nhiều câu đố 8 chữ
- Mỗi giải pháp tối ưu cho một trường hợp vấn đề cung cấp ví dụ từ đó có thể học được $h(\cdot)$
- Học thuật toán
 - Mạng lưới thần kinh
 - Cây quyết định
 - Học quy nạp
 - .

