



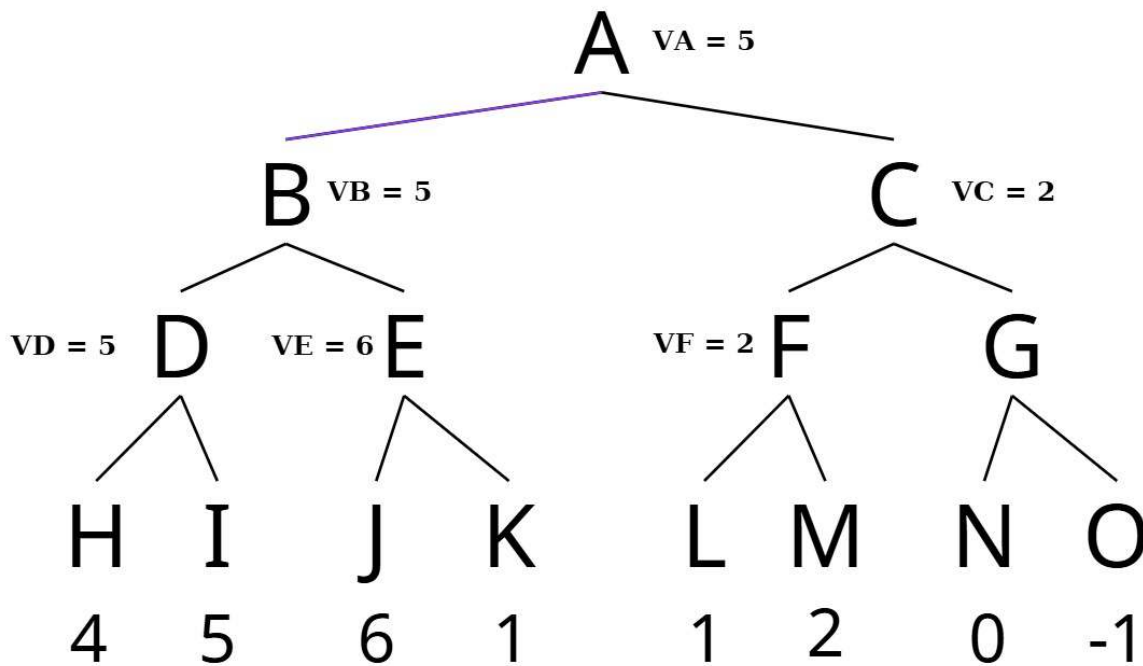
TL AI - tai lieu on tap thi AI

VKU_Trí tuệ nhân tạo (Trường Đại học Công nghệ Thông tin và Truyền thông Việt - Hàn)



Scan to open on Studeersnel

CẮT TỈA AB



$VA = -\infty$; $aA = -\infty$; $bA = +\infty$
 $VB = +\infty$; $aB = -\infty$; $bB = +\infty$
 $VD = -\infty$; $aD = -\infty$; $bD = +\infty$

$VD = \max(VD; VH) = \max(-\infty; 4) = 4$
 $aD = \max(VD; aD) = \max(4; -\infty) = 4$
 $aD < bD (4 < +\infty)$
 $VD = \max(VD; VI) = \max(4; 5) = 5$
 $aD = \max(VD, aD) = \max(5; 4) = 5$

$VB = \min(VB; VD) = \min(+\infty; 5) = 5$
 $bB = \min(VB; bB) = \min(5; +\infty) = 5$

$VE = -\infty$; $aE = -\infty$; $bE = 5$
 $VE = \max(VE; VJ) = \max(-\infty; 6) = 6$
 $aE = \max(VE; aE) = \max(6; -\infty) = 6$
 $aE > bE (6 > 5) \Rightarrow$ cắt bỏ nhánh K

$VB = \min(VB; VE) = \min(5; 6) = 5$
 $bB = \min(VB; bB) = \min(5; 5) = 5$

$VA = \max(VA; VB) = \max(-\infty; 5) = 5$
 $aA = \max(VA; aA) = \max(5; -\infty) = 5$
 $aA < bA (5 < +\infty)$

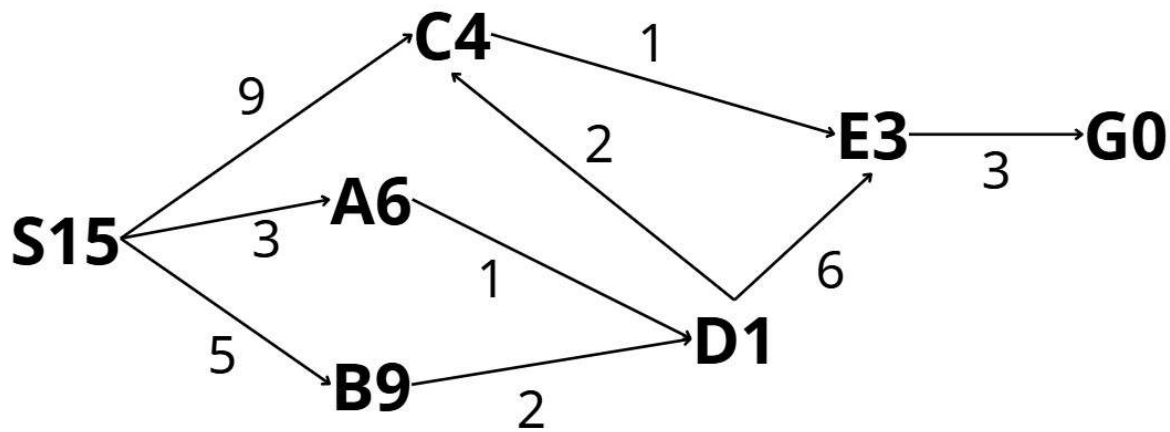
$VC = +\infty$; $aC = 5$; $bC = +\infty$
 $VF = -\infty$; $aF = 5$; $bF = +\infty$
 $VF = \max(VF; VL) = \max(-\infty; 1) = 1$
 $aF = \max(VF; aF) = \max(1; 5) = 5$
 $aF < bF (5 < +\infty)$
 $VF = \max(VF; VM) = \max(1; 2) = 2$
 $aF = \max(VF; aF) = \max(2; 5) = 5$

$VC = \min(VC; VF) = \min(+\infty; 2) = 2$
 $bC = \min(VC; bC) = \min(2; +\infty) = 2$
 $aC > bB (5 > 2) \Rightarrow$ cắt bỏ nhánh G

$VA = \max(VA; VC) = \max(5; 2) = 5$
 $aA = \max(VA; aA) = \max(5; 5) = 5$

Vậy cắt bỏ nhánh K và G

UCS VÀ GREEDY



UCS

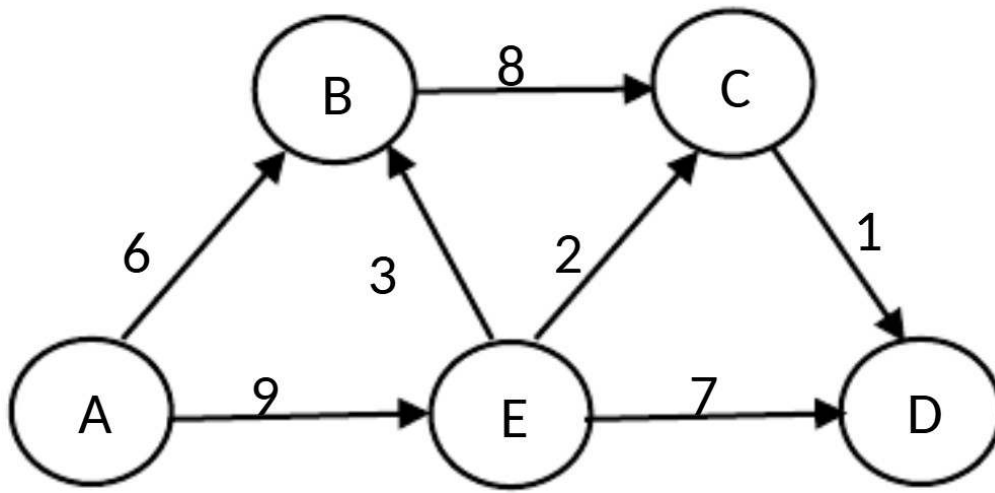
| Bước | Tập đỉnh và chi phí | Đỉnh xét | Tập đỉnh đã xét |
|------|--------------------------|----------|-----------------|
| 1 | { (S;0) } | S | 0 |
| 2 | { (A;3) ; (B;5); (C;9) } | A | {S} |
| 3 | { (D4) (B5) C9 } | D | {S; A} |
| 4 | B5 ; C6 ; E10 | B | S ; A ; D |
| 5 | C6 ; E10 | C | S; A; D; B |
| 6 | E7 | E | S; A; D; B; C |
| 7 | G10 | G | S;A;D;B;C;E |
| 8 | 0 | | SADBCEG |

Vậy Đường đi là S-A-D-C-E-G với chi phí là 10

GREEDY

| Bước | Tập đỉnh và chi phí | Đỉnh xét | Tập đỉnh đã xét |
|------|---------------------|----------|-----------------|
| 1 | { (S;15) } | S | 0 |
| 2 | C4 ; A6; B9 | C | {S} |
| 3 | E3 ; A6 ; B9 | E | S; C |
| 4 | G0; A6; B9 | G | S; C; E |
| 5 | A6; B9 | | S; C; E; G |

Vậy Đường đi là S-C-E-G với chi phí là 13



$h(A) = 8, h(B) = 6, h(C) = 1, h(E) = 5$. Hãy

| Bước | Đỉnh | Kề | $G(v) = G(u) + k(u,v)$ | $F(v) = g(v) + h(v)$ | Open | Closed |
|------|------|--------|---|---|----------|--------|
| 0 | | | | | A8 | |
| 1 | A8 | B6, E5 | $G(B) = G(A) + k(A,B)$ $= 0 + 6 = 6$ $G(E) = 9$ | $fB = gB + hB$ $= 6 + 6 + 12$ $fE = 14$ | B12, E14 | A8 |
| 2 | B12 | C1 | $gC = 14$ | $fC = 15$ | E14, C15 | B12 |
| 3 | E14 | C1, D0 | $gC = 11$ $gD = 16$ | $fC = 12$ $fD = 16$ | C12, D16 | E14 |
| 4 | C12 | D0 | $gD = 12$ | $fD = 12$ | D12 | C12 |
| 5 | D12 | | | | | D |

Vậy đường đi là A-E-C-D vs chi phí là 12

BAYES

-Từ bảng số liệu ta xác định: $n = 13$

-Số bản ghi “play tennis = yes” = 8

-Số bản ghi “play tennis = no” = 5

Ta có: $P(\text{yes}) = 8/13$; $P(\text{no}) = 5/13$

-Xác suất từng phần:

$$P(\text{overcast/yes}) = 2/8 \quad P(\text{overcast/no}) = 2/5$$

$$P(\text{mild/yes}) = 4/8 \quad P(\text{mild/no}) = 2/5$$

$$P(\text{normal/yes}) = 4/8 \quad P(\text{Normal/no}) = 2/5$$

$$P(\text{weak/yes}) = 3/8 \quad P(\text{w/n}) = 3/5$$

Xác suất tổng thể cho từng lớp

$$P(X/\text{yes}) = P(\text{yes}) * P(\text{overcast/yes}) * P(\text{mild/yes}) * P(\text{normal/yes}) * P(\text{weak/yes}) =$$

$$P(X/\text{no}) = P(\text{no}) * P(\text{overcast/yes}) * P(\text{mild/yes}) * P(\text{normal/yes}) * P(\text{weak/yes}) =$$

So sánh tỉ lệ

HQTT

Ta có ptr hqtt: $y^{\wedge} = 0_0 + 0_1 * x$

Trong đó x là diện tích

Y^{\wedge} là gtri dự đoán

0_0 và 0_1 là các hệ số tính từ dữ liệu cho sẵn

Ta có $x - (x \text{ tb}) = 72 \text{ m}^2$

$y - (y \text{ tb}) = 1780 \text{ vpn}$

$$0_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = (50-72)(1200-1780) + \dots / (50-72)^2 + \dots =$$

$$0_0 = (y -) - 0_1 * (x -) =$$

$$\Rightarrow \text{Ptr sẽ là : } Y^{\wedge} = 484.93 + 17 x$$

Tính ẩn số thay vào:

Lý thuyết

Unsupervised Learning (Học không giám sát) là một nhánh của Machine Learning, trong đó mô hình được huấn luyện trên dữ liệu mà không có nhãn (label) đi kèm. Mục tiêu của Unsupervised Learning là tìm kiếm các cấu trúc, mô hình, hoặc mối quan hệ tiềm ẩn trong dữ liệu. Thay vì dự đoán một giá trị đầu ra cụ thể, nó tập trung vào việc khám phá các mẫu (patterns) hoặc nhóm (clusters) trong dữ liệu.

- Đặc điểm chính của Unsupervised Learning:

- Không có nhãn (label): Dữ liệu chỉ chứa các đặc trưng (features), không có giá trị mục tiêu (target).
- Mục tiêu: Tìm các nhóm dữ liệu tương tự nhau (clustering).

Giảm số chiều dữ liệu để dễ dàng phân tích (dimensionality reduction).

Phát hiện bất thường (anomaly detection).

Tìm kiếm các mối quan hệ (association) giữa các dữ liệu.

2 bài toán thuộc Unsupervised Learning

1. Clustering (Phân cụm)

Mục tiêu: Chia dữ liệu thành các nhóm (clusters) sao cho các điểm dữ liệu trong cùng một nhóm giống nhau hơn so với các điểm ở nhóm khác.

Ví dụ:

Phân cụm khách hàng trong kinh doanh để cá nhân hóa chiến lược marketing.

Phân loại gen di truyền hoặc tế bào trong sinh học.

2. Dimensionality Reduction (Giảm số chiều dữ liệu)

Mục tiêu: Giảm số chiều của dữ liệu trong khi vẫn bảo toàn tối đa thông tin quan trọng, giúp dễ dàng trực quan hóa hoặc phân tích.

Ví dụ:

Sử dụng PCA (Principal Component Analysis) để giảm chiều dữ liệu trong xử lý ảnh.

Giảm số chiều dữ liệu để tăng tốc độ huấn luyện trong các bài toán lớn.

Học giám sát (Supervised Learning) là một phương pháp trong Machine Learning, nơi mô hình được huấn luyện trên một tập dữ liệu **có nhãn (labeled data)**. Điều này có nghĩa là mỗi dữ liệu đầu vào (input) đều đi kèm với một giá trị đầu ra (output) mong muốn. Mục tiêu của mô hình là học cách ánh xạ từ đầu vào sang đầu ra để có thể dự đoán chính xác các giá trị đầu ra cho dữ liệu mới.

Đặc điểm chính của Học giám sát

Dữ liệu có nhãn: Dữ liệu huấn luyện bao gồm cả đặc trưng (features) và nhãn (labels).

Mục tiêu:

Dự đoán kết quả đầu ra dựa trên dữ liệu đầu vào.

Tối thiểu hóa sai số giữa dự đoán của mô hình và nhãn thực tế.

Phân loại bài toán:

Classification (Phân loại): Dự đoán nhãn đầu ra rời rạc (discrete labels).

Regression (Hồi quy): Dự đoán giá trị đầu ra liên tục (continuous values).

2 bài toán phổ biến thuộc Học giám sát

1. Phân loại (Classification)

Mục tiêu: Phân loại dữ liệu vào các nhóm hoặc lớp (classes) khác nhau.

Ví dụ: Phân loại email thành "spam" hoặc "không spam".

Xác định hình ảnh là "chó", "mèo", hay "chim".

2. Hồi quy (Regression)

Mục tiêu: Dự đoán giá trị liên tục dựa trên dữ liệu đầu vào.

Ví dụ: Dự đoán giá nhà dựa trên diện tích, vị trí, số phòng, v.v.

Dự đoán nhiệt độ trong ngày dựa trên dữ liệu thời tiết.

| Đặc điểm | Học giám sát | Học không giám sát |
|----------------|---|--|
| Dữ liệu | Có nhãn (labeled data) | Không có nhãn (unlabeled data) |
| Mục tiêu | Dự đoán đầu ra dựa trên dữ liệu đầu vào | Tìm cấu trúc ẩn trong dữ liệu |
| Ví dụ bài toán | Phân loại (Classification), Hồi quy (Regression) | Phân cụm (Clustering), Giảm số chiều (Dimensionality Reduction) |

Hồi quy (Regression) là một bài toán trong học giám sát (Supervised Learning), nơi mục tiêu là dự đoán một giá trị đầu ra liên tục (continuous value) dựa trên dữ liệu đầu vào. Hồi quy được sử dụng để tìm mối quan hệ giữa các biến đầu vào (features) và biến đầu ra (target).

Mục tiêu của Hồi quy

- Xây dựng một mô hình toán học để dự đoán giá trị của một biến liên tục.
- Tối thiểu hóa sai số (error) giữa các giá trị dự đoán và giá trị thực tế.

Ví dụ về các bài toán Hồi quy

Dự đoán giá nhà

Đầu vào: Diện tích, số phòng, vị trí, năm xây dựng.

Đầu ra: Giá nhà (đơn vị tiền tệ).

Dự đoán doanh thu

Đầu vào: Lượng khách hàng, giá sản phẩm, chi phí quảng cáo.

Đầu ra: Doanh thu (đơn vị tiền tệ).

Các thuật toán hồi quy phổ biến: Linear Regression, Polynomial Regression, Ridge Regression và Lasso Regression, Decision Tree Regression, ...

Phân lớp (Classification) là một bài toán trong học giám sát (Supervised Learning), nơi mục tiêu là dự đoán một nhãn (label) hoặc lớp (class) cho dữ liệu đầu vào. Giá trị đầu ra của bài toán phân lớp là rời rạc (discrete), tức là thuộc một trong các nhóm hoặc danh mục xác định trước.

Mục tiêu của Phân lớp

Xây dựng một mô hình để gán nhãn đúng cho dữ liệu đầu vào.

Tối đa hóa độ chính xác (accuracy) của dự đoán.

Ví dụ về các bài toán Phân lớp

Phân loại email:

Đầu vào: Nội dung email, tiêu đề.

Đầu ra: "Spam" hoặc "Không spam".

Chẩn đoán bệnh:

Đầu vào: Các chỉ số sức khỏe (nhịp tim, huyết áp, kết quả xét nghiệm).

Đầu ra: "Bệnh tiểu đường", "Không bệnh".

Các thuật toán phân lớp phổ biến: Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors..

Perceptron là gì?

Perceptron là một thuật toán học máy cơ bản trong học giám sát (supervised learning), được giới thiệu bởi Frank Rosenblatt vào năm 1958. Nó là một trong những mô hình nền tảng của mạng nơ-ron nhân tạo (Artificial Neural Networks), được thiết kế để mô phỏng cách thức hoạt động của nơ-ron sinh học.

Perceptron là một mô hình đơn giản dùng để phân loại dữ liệu tuyến tính thành hai lớp (nhị phân). Nó dựa trên ý tưởng học cách kết hợp các trọng số với các đặc trưng đầu vào để đưa ra quyết định.

Cấu trúc của Perceptron

Một Perceptron bao gồm các thành phần chính sau:

1. Đầu vào (x): Tập hợp các đặc trưng (features) của dữ liệu.

Ví dụ: $x=[x_1, x_2, \dots, x_n]$

2. Trọng số (w): Bộ trọng số gắn với mỗi đặc trưng.

Ví dụ: $w=[w_1, w_2, \dots, w_n]$

3. Hàm tính tổng (Weighted Sum): Tính tổng có trọng số của đầu vào.

$$z = \sum_{i=1}^n w_i x_i + b$$

$$z = \sum_{i=1}^n w_i x_i + b$$

Trong đó:

b: Là bias (hệ số chặn), giúp mô hình dịch chuyển đường quyết định.

4. Hàm kích hoạt (Activation Function): Xử lý giá trị tổng (z) để đưa ra quyết định.

Thường sử dụng hàm kích hoạt đơn giản:

$$y = \begin{cases} 1, & \text{nếu } z \geq 0 \\ 0, & \text{nếu } z < 0 \end{cases}$$

Kết quả đầu ra y là nhãn dự đoán (0 hoặc 1).

So sánh các thuật toán tìm đường đi ngắn nhất: UCS, Greedy, và A*

1. Thuật toán Uniform-Cost Search (UCS)

Đặc điểm chính:

UCS là một thuật toán tìm kiếm dựa trên chi phí thực tế (actual cost) từ điểm bắt đầu đến các đỉnh.

Nó là một phiên bản tổng quát của thuật toán Dijkstra.

Ưu tiên mở rộng các đỉnh có chi phí tích lũy thấp nhất.

Hàm đánh giá (Evaluation Function):

$$f(n)=g(n)$$

$g(n)$: Chi phí từ điểm xuất phát đến đỉnh n .

Thuộc tính:

Tối ưu: Có, nếu chi phí các cạnh không âm (non-negative edge costs).

Hoàn chỉnh: Có, nếu không gian trạng thái là hữu hạn và các chi phí cạnh không âm.

Hiệu quả: Tìm được đường đi ngắn nhất, nhưng không sử dụng thông tin về khoảng cách ước lượng đến đích.

Ưu điểm:

Đảm bảo tìm được đường đi ngắn nhất.

Phù hợp cho bài toán mà chi phí đi lại giữa các đỉnh là trọng tâm.

Nhược điểm:

Hiệu suất kém hơn khi không gian tìm kiếm lớn vì không sử dụng thông tin về đích.

Có thể mở rộng quá nhiều đỉnh không cần thiết.

2. Thuật toán Greedy Best-First Search

Đặc điểm chính:

Greedy Best-First Search là thuật toán dựa trên ước lượng (heuristic) để dẫn đường cho tìm kiếm.

Nó ưu tiên mở rộng các đỉnh gần đích nhất theo ước lượng.

Hàm đánh giá (Evaluation Function):

$$f(n)=h(n)$$

$h(n)$: Ước lượng chi phí từ đỉnh n đến đích (heuristic).

Thuộc tính:

Tối ưu: Không đảm bảo tìm được đường đi ngắn nhất.

Hoàn chỉnh: Không, nếu không gian tìm kiếm vô hạn hoặc heuristic không đảm bảo tính chính xác.

Hiệu quả: Có thể tìm đường đi nhanh hơn UCS nếu heuristic tốt, nhưng không đảm bảo tối ưu.

Ưu điểm:

Tốc độ nhanh hơn UCS khi heuristic tốt.

Thích hợp để tìm đường trong không gian lớn nếu không cần đường đi ngắn nhất.

Nhược điểm:

Không đảm bảo tìm đường đi ngắn nhất.

Có thể rơi vào bẫy heuristic xấu, dẫn đến tìm kiếm không hiệu quả.

3. Thuật toán A*

Đặc điểm chính:

A* là sự kết hợp giữa UCS và Greedy Best-First Search. Nó sử dụng cả chi phí thực tế và ước lượng heuristic để dẫn đường.

Mở rộng các đỉnh có tổng chi phí tích lũy và ước lượng đến đích thấp nhất.

Hàm đánh giá (Evaluation Function):

$$f(n) = g(n) + h(n)$$

$g(n)$: Chi phí từ điểm xuất phát đến đỉnh n .

$h(n)$: Ước lượng chi phí từ đỉnh n đến đích.

Thuộc tính:

Tối ưu: Có, nếu heuristic là admissible (không đánh giá quá cao chi phí thực tế) và consistent (tuân thủ bất đẳng thức tam giác).

Hoàn chỉnh: Có, nếu không gian tìm kiếm là hữu hạn và heuristic là admissible.

Hiệu quả: Tìm đường nhanh hơn UCS nhờ sử dụng heuristic để giảm số lượng đỉnh mở rộng.

Ưu điểm:

Đảm bảo tìm đường đi ngắn nhất nếu heuristic tốt.

Hiệu quả hơn UCS khi kết hợp giữa chi phí thực tế và ước lượng.

Nhược điểm:

Hiệu suất phụ thuộc vào chất lượng của heuristic.

Chi phí tính toán heuristic có thể cao với các bài toán phức tạp.

So sánh tổng quan

| Thuật toán | Hàm đánh giá | Tối ưu | Hoàn chỉnh | Ưu tiên mở rộng | Ưu điểm | Nhược điểm |
|------------|------------------|--------|------------|-------------------------------------|-------------------------------------|--|
| UCS | $f(n)=g(n)$ | Có | Có | Đỉnh có chi phí tích lũy thấp nhất | Đảm bảo đường đi ngắn nhất | Chậm khi không gian lớn |
| Greedy | $f(n)=h(n)$ | Không | Không | Đỉnh gần đích nhất (theo ước lượng) | Tìm đường nhanh khi heuristic tốt | Không đảm bảo tối ưu, dễ kẹt ở heuristic xấu |
| A* | $f(n)=g(n)+h(n)$ | Có | Có | Đỉnh có tổng $g(n)+h(n)$ thấp nhất | Hiệu quả, tìm đường nhanh và tối ưu | Phụ thuộc vào chất lượng của heuristic |

so sánh sự khác nhau giữa greedy và A*

| Tiêu chí | Greedy | A* |
|-------------------|----------------------|--------------------------------|
| Hàm đánh giá | $f(n) = h(n)$ | $f(n) = g(n) + h(n)$ |
| Tính chính xác | Không đảm bảo tối ưu | Đảm bảo tối ưu nếu $h(n)$ đúng |
| Tốc độ | Thường nhanh hơn | Chậm hơn |
| Sử dụng heuristic | Chỉ dựa vào $h(n)$ | Dựa cả $g(n)$ và $h(n)$ |
| Ứng dụng | Tìm kiếm nhanh | Tìm kiếm tối ưu |

| Tiêu chí | Tìm kiếm theo chiều rộng (BFS) | Tìm kiếm theo chiều sâu (DFS) | Tìm kiếm chiều sâu lặp (Iterative Deepening) |
|---------------------------------|--|--|--|
| Chiến lược tìm kiếm | Duyệt theo từng mức (layer) của cây/đồ thị từ gốc đến đích. | Duyệt sâu tối đa vào một nhánh trước, sau đó quay lại. | Kết hợp BFS và DFS: Duyệt sâu từng mức tăng dần theo chiều sâu. |
| Đảm bảo tìm được lời giải? | Có, nếu có lời giải. | Có, nếu có lời giải và không có chu trình vô hạn. | Có, nếu có lời giải. |
| Đảm bảo tối ưu? | Có, nếu chi phí mỗi bước là bằng nhau. | Không, không đảm bảo tối ưu. | Có, nếu chi phí mỗi bước là bằng nhau. |
| Tính đầy đủ (Completeness) | Có, nếu không gian trạng thái hữu hạn. | Có thể không, nếu không xử lý chu trình. | Có, nếu không gian trạng thái hữu hạn. |
| Bộ nhớ cần sử dụng | Cao, vì phải lưu trữ tất cả các nút trong hàng đợi. | Thấp hơn BFS, chỉ cần lưu đường đi hiện tại. | Trung bình, thấp hơn BFS nhưng cao hơn DFS (do lặp lại). |
| Tốc độ (Thời gian) | Chậm hơn DFS (vì duyệt toàn bộ mức). | Nhanh hơn BFS trong trường hợp cây rộng, nhưng có thể mất nhiều thời gian nếu nhánh không chứa lời giải. | Chậm hơn BFS do phải duyệt lại, nhưng hiệu quả hơn DFS không gian lớn. |
| Tìm kiếm trong không gian lớn | Không hiệu quả (do bộ nhớ lớn). | Hiệu quả hơn BFS, nhưng có thể không tìm được lời giải. | Hiệu quả hơn BFS và DFS trong không gian lớn. |
| Áp dụng cho đồ thị có chu trình | Cần kiểm tra và loại bỏ các chu trình. | Cần kiểm tra và loại bỏ các chu trình. | Tự động xử lý các lần lặp. |
| Ưu điểm | - Đảm bảo tìm được lời giải tối ưu (nếu có).- Tốt cho bài toán có lời giải nông. | - Tiêu tốn ít bộ nhớ.- Tốt cho các bài toán có lời giải sâu. | - Đảm bảo cả tính đầy đủ.- Cân bằng giữa BFS và DFS. |
| Nhược điểm | - Tiêu tốn nhiều bộ nhớ. Không phù hợp với không gian trạng thái lớn. | - Không đảm bảo tối ưu. Có thể rơi vào vòng lặp vô hạn. | - Thời gian xử lý lâu hơn DFS do phải duyệt lại nhiều lần. |