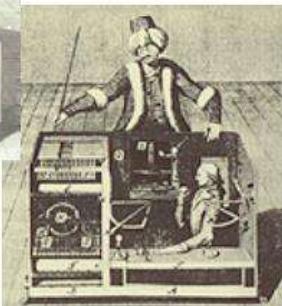




Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

Bộ môn Công Nghệ Phần Mềm  
Khoa Công Nghệ Thông Tin

Nha trang - 2012



## NỘI DUNG

- GIỚI THIỆU
- THUẬT TOÁN, THUẬT GIẢI
- GIẢI THUẬT DI TRUYỀN
- BIỂU DIỄN VÀ SUY LUẬN TRI THỨC
- LOGIC MỆNH ĐỀ LOGIC VỊ TỪ
- MÁY HỌC
- MẠNG NEURAL
- SVM
- HMM



## GIỚI THIỆU

- AI là lĩnh vực của Công nghệ thông tin, có chức năng nghiên cứu và tạo ra các chương trình mô phỏng hoạt động tư duy của con người.
- Trí tuệ nhân tạo nhằm tạo ra “Máy người”?
- Mục tiêu
- Xây dựng lý thuyết về thông minh để giải thích các hoạt động thông minh
- Tìm hiểu cơ chế sự thông minh của con người
  - Cơ chế lưu trữ tri thức
  - Cơ chế khai thác tri thức
- Xây dựng cơ chế hiện thực sự thông minh
- Áp dụng các hiểu biết này vào các máy móc phục vụ con người



## GIỚI THIỆU

### ĐỐI TƯỢNG NGHIÊN CỨU CỦA AI

- AI là ngành nghiên cứu về cách hành xử thông minh (intelligent behaviour) bao gồm: thu thập, lưu trữ tri thức, suy luận, hoạt động và kỹ năng.
- Đối tượng nghiên cứu là các “hành xử thông minh” chứ không phải là “sự thông minh”.
- Giải quyết bài toán bằng AI là tìm cách biểu diễn tri thức, tìm cách vận dụng tri thức để giải quyết vấn đề và tìm cách bổ sung tri thức bằng cách “phát hiện” tri thức từ những thông tin sẵn có (máy học)



## GIỚI THIỆU

### LỊCH SỬ PHÁT TRIỂN CỦA AI GIAI ĐOẠN CỔ ĐIỂN

Giai đoạn cổ điển (1950 – 1965)

Có 2 kỹ thuật tìm kiếm cơ bản:

- ▣ Kỹ thuật generate and test : chỉ tìm được 1 đáp án/ chưa chắc tối ưu.
- ▣ Kỹ thuật Exhaustive search (vét cạn): Tìm tất cả các nghiệm, chọn lựa phương án tốt nhất



## GIỚI THIỆU

### LỊCH SỬ PHÁT TRIỂN CỦA AI GIAI ĐOẠN VIỄN VÔNG

- Đây là giai đoạn phát triển với tham vọng làm cho máy hiểu được con người qua ngôn ngữ tự nhiên.
- Các công trình nghiên cứu tập trung vào việc biểu diễn tri thức và phương thức giao tiếp giữa người và máy bằng ngôn ngữ tự nhiên.
- Kết quả không mấy khả quan nhưng cũng tìm ra được các phương thức biểu diễn tri thức vẫn còn được dùng đến ngày nay tuy chưa thật tốt như:

- ▣ Semantic Network (mạng ngữ nghĩa)
- ▣ Conceptual graph (đồ thị khái niệm)
- ▣ Frame (khung)
- ▣ Script (kịch bản)

Vấp phải trở ngại về năng lực  
của máy tính



## GIỚI THIỆU

### LỊCH SỬ PHÁT TRIỂN CỦA AI GIAI ĐOẠN HIỆN ĐẠI

Giai đoạn hiện đại (từ 1975)

- Xác định lại mục tiêu mang tính thực tiễn hơn của AI:
  - Tìm ra **lời giải tốt nhất** trong khoảng thời gian chấp nhận được.
  - Không cầu toàn tìm ra **lời giải tối ưu**
- Tinh thần HEURISTIC ra đời và được áp dụng mạnh mẽ để khắc phục bùng nổ tổ hợp.
- Khẳng định vai trò của tri thức đồng thời xác định 2 trở ngại lớn là biểu diễn tri thức và bùng nổ tổ hợp.
- Nêu cao vai trò của Heuristic nhưng cũng khẳng định tính khó khăn trong đánh giá heuristic.



## GIỚI THIỆU

### CÁC LĨNH VỰC ỨNG DỤNG

- Game Playing: Tìm kiếm / Heuristic
- Automatic reasoning & Theorem proving: Tìm kiếm / Heuristic
- Expert System: là hướng phát triển mạnh mẽ nhất và có giá trị ứng dụng cao nhất.
- Planning & Robotic: các hệ thống dự báo, tự động hóa
- Machine learning: Trang bị khả năng học tập để giải quyết vấn đề kho tri thức:
  - Supervised : Kiểm soát được tri thức học được. Không tìm ra cái mới.
  - UnSupervised:Tự học, không kiểm soát. Có thể tạo ra tri thức mới nhưng cũng nguy hiểm vì có thể học những điều không mong muốn.



## GIỚI THIỆU

### CÁC LĨNH VỰC ỨNG DỤNG

- Natural Language Understanding & Semantic modelling: Không được phát triển mạnh do mức độ phức tạp của bài toán cả về tri thức & khả năng suy luận.
- Modeling Human performance: Nghiên cứu cơ chế tổ chức trí tuệ của con người để áp dụng cho máy.
- Language and Environment for AI: Phát triển công cụ và môi trường để xây dựng các ứng dụng AI.
- Neural network / Parallel Distributed processing: giải quyết vấn đề năng lực tính toán và tốc độ tính toán bằng kỹ thuật song song và mô phỏng mạng thần kinh của con người

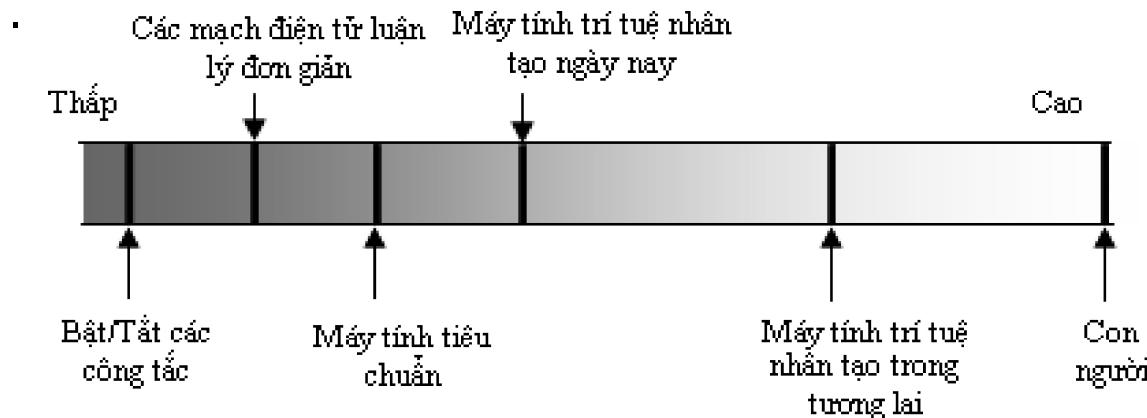


## GIỚI THIỆU

### ỨNG DỤNG AI

Mô hình ứng dụng AI hiện tại:

**AI = Presentation & Search**



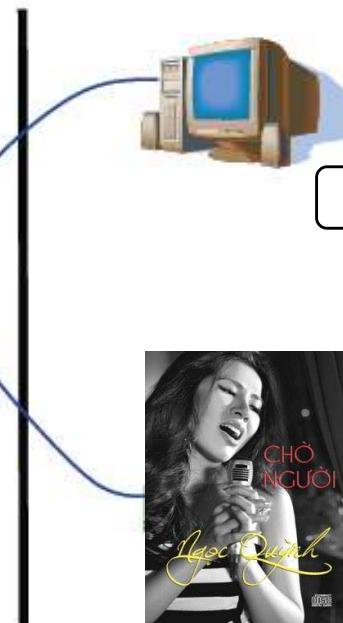


# GIỚI THIỆU

## Turing Test

Judge types questions to both.

- Computer claims to be human.
- (So does human, btw!)



My song



Google



If judge cannot tell which entity is human, we say computer has passed the test and is "intelligent."

Artificial Intelligence

Nguyễn Đình Cường 11



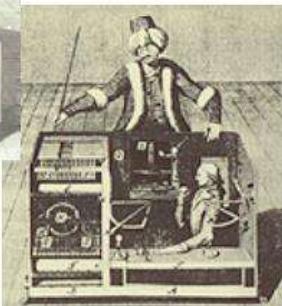
Artificial Intelligence

Nguyễn Đình Cường 12



Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

Bộ môn Công Nghệ Phần Mềm  
Khoa Công Nghệ Thông Tin

Nha trang - 2012



## THUẬT TOÁN - THUẬT GIẢI

- Trong quá trình nghiên cứu giải quyết các vấn đề – bài toán, người ta đã đưa ra những nhận xét như sau:
  - Có nhiều bài toán cho đến nay vẫn chưa tìm ra một cách giải theo kiểu thuật toán và cũng không biết là có tồn tại thuật toán hay không.
  - Có nhiều bài toán đã có thuật toán để giải nhưng không chấp nhận được vì thời gian giải theo thuật toán đó quá lớn hoặc các điều kiện cho thuật toán khó đáp ứng.
  - Có những bài toán được giải theo những cách giải vi phạm thuật toán nhưng vẫn chấp nhận được.



## THUẬT TOÁN - THUẬT GIẢI

- Thuật giải Heuristic là một sự mở rộng khái niệm thuật toán. Nó thể hiện cách giải bài toán với các đặc tính sau:
  - Thường tìm được lời giải tốt (nhưng không chắc là lời giải tốt nhất)
  - Giải bài toán theo thuật giải Heuristic thường dễ dàng và nhanh chóng đưa ra kết quả hơn so với giải thuật tối ưu, vì vậy chi phí thấp hơn.
  - Thuật giải Heuristic thường thể hiện khá tự nhiên, gần gũi với cách suy nghĩ và hành động của con người



## THUẬT TOÁN - THUẬT GIẢI

Một số nguyên lý cơ bản như sau

### Nguyên lý vét cạn thông minh

Trong một 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 (Greedy)

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 của từng bước (hay từng giai đoạn) 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.

### Hàm Heuristic

Trong việc xây dựng các thuật giải Heuristic, người ta thường dùng các hàm Heuristic. Đó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào trạng thái hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này, ta có thể chọn được cách hành động tương đối hợp lý trong từng bước của thuật giải.



# THUẬT TOÁN - THUẬT GIẢI

## CÁC BÀI TOÁN

- Đổi tiền (Vét cạn và Heuristic)
- Tìm kiếm chiềú rộng và sâu
- Tic tac toe.
- Đong dầu.
- Bài toán TSP
- Tô màu bản đồ
- Tổ chức Hội nghị
- 8,16 puzzle
- Cờ vua, cờ tướng
- Người nông dân qua sông.
- Con thỏ và con cáo
- Con khỉ và nải chuối



# THUẬT TOÁN - THUẬT GIẢI

## CÁC BÀI TOÁN

- Trò chơi Nim
- Trò chơi Ô quan
- Giải bài toán tam giác tự động
- Tính tích phân bất định
- Giải phương trình bậc cao bằng giải thuật di truyền
- Tính giá trị lớn nhất của hàm nhiều biến bằng giải thuật di truyền
- Sắp xếp thời khóa biểu bằng giải thuật di truyền
- Cài đặt thuật giải Robinson chứng minh mệnh đề
- Cài đặt thuật giải Vương Hạo chứng minh mệnh đề
- Cài đặt thuật giải Quinland trong máy học
- Cài đặt thuật giải học theo Entropy



# THUẬT TOÁN - THUẬT GIẢI

## CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

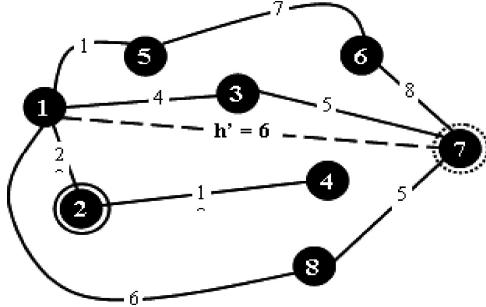
### Cấu trúc chung của bài toán tìm kiếm

"Xuất phát từ một đỉnh của một đồ thị, tìm đường đi hiệu quả nhất đến một đỉnh nào đó".

Một phát biểu khác thường gặp của dạng bài toán này là :

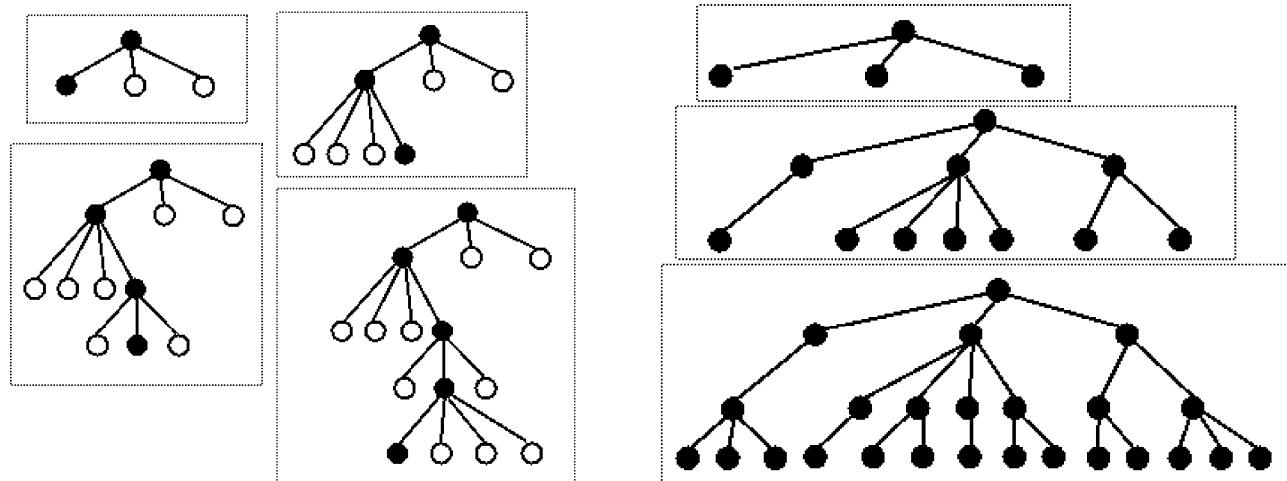
Cho trước hai trạng thái  $T_0$  và  $T_G$  hãy xây dựng chuỗi trạng thái  $T_0, T_1, T_2, \dots, T_{n-1}$ ,

$T_n = T_G$  sao cho  $\sum_{i=1}^n \text{cost}(T_{i-1}, T_i)$  thỏa mãn một điều kiện cho trước (thường là nhỏ nhất).



# THUẬT TOÁN - THUẬT GIẢI

## Tìm kiếm chiều sâu và tìm kiếm chiều rộng



Tìm kiếm chiều sâu (Depth-First Search)

Tìm kiếm chiều rộng (Breadth-First Search )



## THUẬT TOÁN - THUẬT GIẢI

	Chiều sâu	Chiều rộng
Tính hiệu quả	Hiệu quả khi lời giải nằm sâu trong cây tìm kiếm và có một phương án chọn hướng đi chính xác. Hiệu quả của chiến lược phụ thuộc vào phương án chọn hướng đi. Phương án càng kém hiệu quả thì hiệu quả của chiến lược càng giảm. Thuận lợi khi muốn tìm chỉ một lời giải.	Hiệu quả khi lời giải nằm gần gốc của cây tìm kiếm. Hiệu quả của chiến lược phụ thuộc vào độ sâu của lời giải. Lời giải càng xa gốc thì hiệu quả của chiến lược càng giảm. Thuận lợi khi muốn tìm nhiều lời giải.
Lượng bộ nhớ sử dụng để lưu trữ các trạng thái	Chỉ lưu lại các trạng thái chưa xét đến.	Phải lưu toàn bộ các trạng thái.
Trường hợp xấu nhất	Vét cạn toàn bộ	Vét cạn toàn bộ.
Trường hợp tốt nhất	Phương án chọn hướng đi <i>tuyệt đối</i> chính xác. Lời giải được xác định một cách trực tiếp.	Lời giải được xác định khi mở nút đầu tiên.

Artificial Intelligence

Nguyễn Đình Cường 21



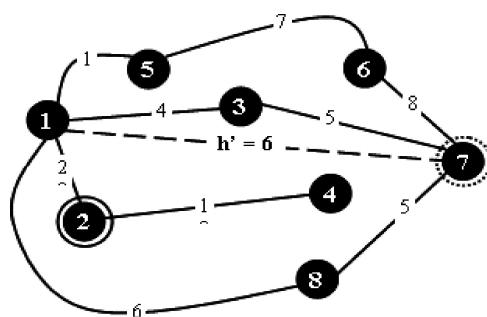
## THUẬT TOÁN - THUẬT GIẢI

### Tìm kiếm leo đồi

- Tìm kiếm leo đồi theo đúng nghĩa, nói chung, thực chất chỉ là một trường hợp đặc biệt của tìm kiếm theo chiều sâu nhưng không thể quay lui. Trong tìm kiếm leo đồi, việc lựa chọn trạng thái tiếp theo được quyết định dựa trên một hàm Heuristic

### Hàm Heuristic là gì

- Thuật ngữ "hàm Heuristic" muốn nói lên điều gì? Chẳng có gì ghê gớm. Bạn đã quen với nó rồi! Đó đơn giản chỉ là một **ước lượng về khả năng dẫn đến lời giải** tính từ trạng thái đó (*khoảng cách* giữa trạng thái hiện tại và trạng thái đích).

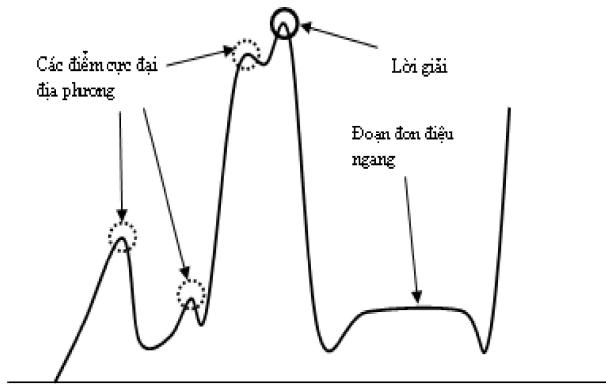


Artificial Intelligence

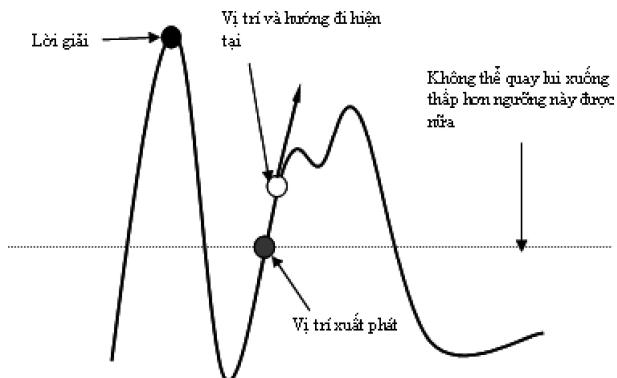
Nguyễn Đình Cường 22



## THUẬT TOÁN - THUẬT GIẢI



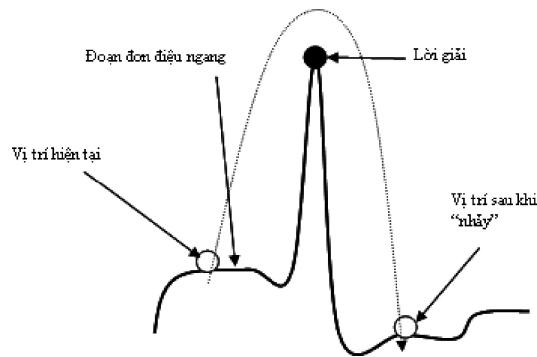
Hình : Các tình huống khó khăn cho tìm kiếm leo đèo



Hình : Một trường hợp thất bại của leo đèo kết hợp quay lui



## THUẬT TOÁN - THUẬT GIẢI



Hình Một trường hợp khó khăn cho phương án "nhảy vọt"

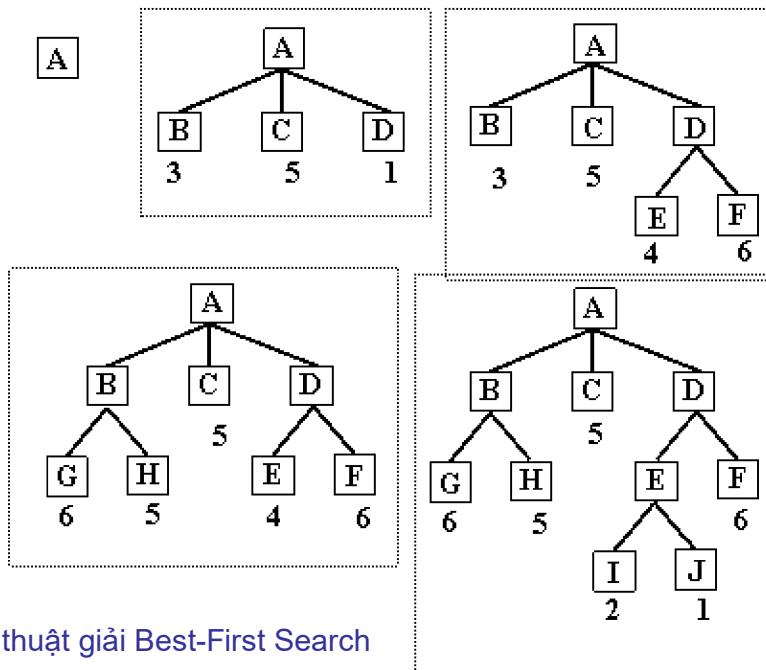
Hiệu quả của cả hai thuật giải leo đồi đơn giản và leo đồi dốc đứng phụ thuộc vào :

- Chất lượng của hàm Heuristic.
- Đặc điểm của không gian trạng thái.
- Trạng thái khởi đầu



## THUẬT TOÁN - THUẬT GIẢI

Tìm kiếm ưu tiên tối ưu (best-first search)



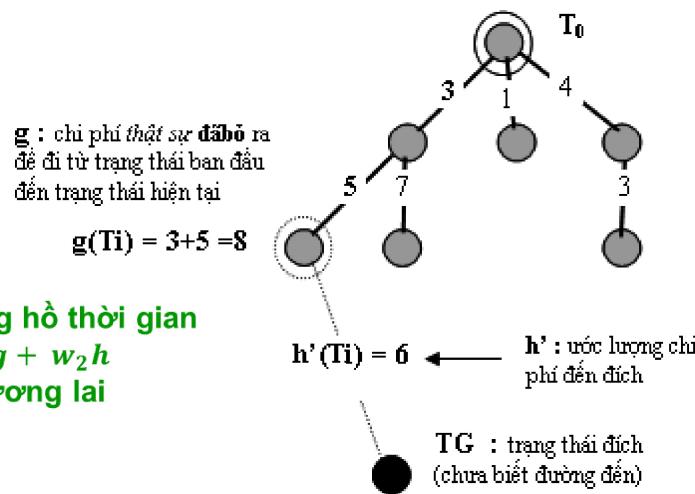
Hình Minh họa thuật giải Best-First Search



## THUẬT TOÁN - THUẬT GIẢI

- BFS khá đơn giản. Tuy vậy, trên thực tế, cũng như tìm kiếm chiều sâu và chiều rộng, hiếm khi ta dùng BFS một cách trực tiếp. Thông thường, người ta thường dùng các phiên bản của BFS là AT, AKT và A\*

- Thông tin về quá khứ và tương lai**



Câu chuyện chiếc đồng hồ thời gian

$$f = w_1 g + w_2 h$$

Quá khứ, hiện tại và tương lai

Hình Phân biệt khái niệm g và h'



# THUẬT TOÁN - THUẬT GIẢI

## Thuật giải AT

Thuật giải AT là một phương pháp tìm kiếm theo kiểu BFS với độ tốt của nút là giá trị hàm **g** tổng chiều dài con đường đã đi từ trạng thái bắt đầu đến trạng thái hiện tại.

## Thuật giải AT

**1.** Đặt **OPEN** chứa trạng thái khởi đầu.

**2.** Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện :

**2.a.** Chọn trạng thái ( $T_{max}$ ) có **giá trị g nhỏ nhất** trong OPEN (và xóa  $T_{max}$  khỏi OPEN)

**2.b.** Nếu  $T_{max}$  là trạng thái kết thúc thì thoát.

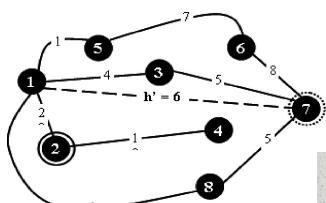
**2.c.** Ngược lại, tạo ra các trạng thái kế tiếp  $T_k$  có thể có từ trạng thái  $T_{max}$ . Đối với mỗi trạng thái kế tiếp  $T_k$  thực hiện :

$$g(T_k) = g(T_{max}) + cost(T_{max}, T_k);$$

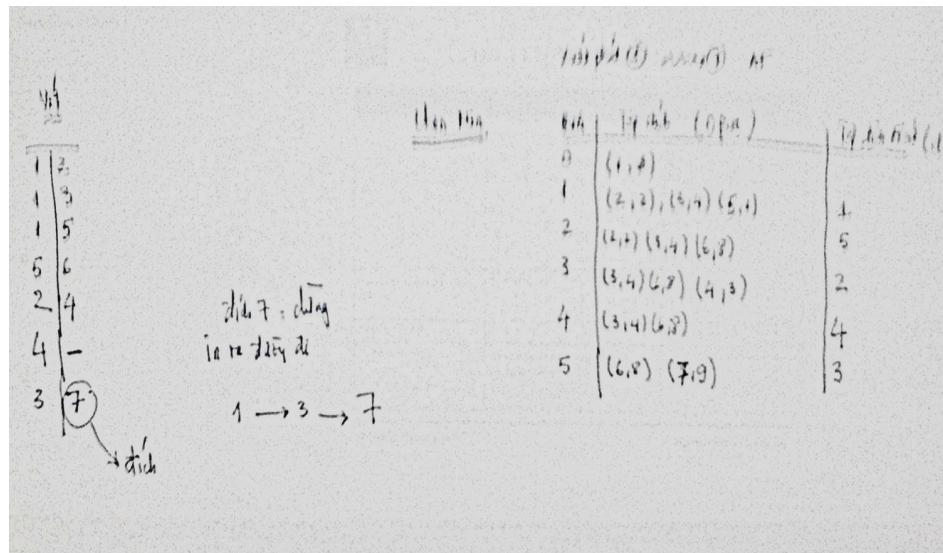
Thêm  $T_k$  vào OPEN.



# THUẬT TOÁN - THUẬT GIẢI



**Tìm đường đi ngắn nhất từ đỉnh 1 đến đỉnh 7**  
**Thuật giải AT**





## THUẬT TOÁN - THUẬT GIẢI

### Thuật giải AKT (*Algorithm for Knowledgeable Tree Search*)

Thuật giải AKT mở rộng AT bằng cách sử dụng thêm thông tin ước lượng  $h'$ . Độ tốt của một trạng thái  $f$  là tổng của hai hàm  $g$  và  $h'$ .

#### Thuật giải AKT

1. Đặt **OPEN** chứa trạng thái khởi đầu.
2. Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện :
  - 2.a. Chọn trạng thái ( $T_{max}$ ) có **giá trị f nhỏ nhất** trong OPEN (và xóa  $T_{max}$  khỏi OPEN)
  - 2.b. Nếu  $T_{max}$  là trạng thái kết thúc thì thoát.
  - 2.c. Ngược lại, tạo ra các trạng thái kế tiếp  $T_k$  có thể có từ trạng thái  $T_{max}$ . Đối với mỗi trạng thái kế tiếp  $T_k$  thực hiện :

$$g(T_k) = g(T_{max}) + \text{cost}(T_{max}, T_k);$$

#### Tính $h'(T_k)$

$$f(T_k) = g(T_k) + h'(T_k);$$

Thêm  $T_k$  vào OPEN.



## THUẬT TOÁN - THUẬT GIẢI

### Thuật giải A\*

- A\* là một phiên bản đặc biệt của AKT áp dụng cho trường hợp đồ thị.
- Thuật giải A\* có sử dụng thêm tập hợp **CLOSE** để lưu trữ những trường hợp đã được xét đến.
- A\* mở rộng AKT bằng cách bổ sung cách giải quyết trường hợp khi "mở" một nút mà nút này đã có sẵn trong OPEN hoặc CLOSE.
- Khi xét đến một trạng thái  $T_i$  bên cạnh việc lưu trữ 3 giá trị cơ bản  $g$ ,  $h'$ ,  $f'$  để phản ánh độ tốt của trạng thái đó, A\* còn lưu trữ thêm hai thông số sau :
  1. *Trạng thái cha của trạng thái  $T_i$  (ký hiệu là  $Cha(T_i)$ )*
  2. *Danh sách các trạng thái kế tiếp của  $T_i$*



## THUẬT TOÁN - THUẬT GIẢI

**Begin**

open:={s}; close:= $\emptyset$ ;

**While** (open $\neq \emptyset$ ) **do**

**begin**

n:= Retrieve(Open) //sao cho f(n) min.

**if** (n=g) **then** return path từ s đến g

**else begin**

Tạo  $\Gamma(n)$

**for** mỗi nút con m của  $\Gamma(n)$  **do**

**case** m **of**

m  $\notin$  Open và m  $\notin$  Close:

**begin**

Gán giá trị heuristic

cho m

m  $\in$  Open:

**if** đến được m bằng một path ngắn hơn  
**then** Cập nhật lại m trong Open.

m  $\in$  Close

**if** đến được m bằng một path ngắn hơn **then**  
**begin**

Close:=Close-{m}

Open:=Open $\cup$ {m}

**end;**

**end;** /\*end case\*/

Close:=Close $\cup$ {n}

**end;** / while/

return false;

Open:=Open $\cup$ {m}; **End;**

**end;**

**Artificial Intelligence**

**Nguyễn Đình Cường** 31



## THUẬT TOÁN - THUẬT GIẢI

- Hàm lượng giá Heuristic là hàm ước lượng phí tổn để đi từ trạng thái hiện tại đến trạng thái goal.
- Cơ sở để xác định hàm lượng giá là dựa vào tri thức/kinh nghiệm thu thập được.
- Hàm lượng giá cho kết quả đúng (gần thực thể) hay sai (xa giá trị thực) sẽ dẫn đến
- kết quả tìm được tốt hay xấu.
- Không có chuẩn mực cho việc đánh giá một hàm lượng giá Heuristic. Lý do:

Không có cấu trúc chung cho hàm lượng giá

Tính đúng/sai thay đổi liên tục theo từng vấn đề cụ thể

Tính đúng/sai thay đổi theo từng tình huống cụ thể trong một vấn đề

**Có thể dùng nhiều hàm lượng giá khác nhau theo tình huống → cần hàm lượng giá về các hàm lượng giá.**

**Artificial Intelligence**

**Nguyễn Đình Cường** 32



## THUẬT TOÁN - THUẬT GIẢI

Trò đố 8 ô hay 15 ô

<p><b>Trạng thái ban đầu</b></p> <p>- Trò đố 15 ô</p>	<p><b>Trạng thái đích</b></p>																																
<table border="1" style="margin: auto;"> <tr><td>11</td><td>14</td><td>4</td><td>7</td></tr> <tr><td>10</td><td>6</td><td></td><td>5</td></tr> <tr><td>1</td><td>2</td><td>13</td><td>15</td></tr> <tr><td>9</td><td>12</td><td>8</td><td>3</td></tr> </table>	11	14	4	7	10	6		5	1	2	13	15	9	12	8	3	<table border="1" style="margin: auto;"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>12</td><td>13</td><td>14</td><td>5</td></tr> <tr><td>11</td><td></td><td>15</td><td>6</td></tr> <tr><td>10</td><td>9</td><td>8</td><td>7</td></tr> </table>	1	2	3	4	12	13	14	5	11		15	6	10	9	8	7
11	14	4	7																														
10	6		5																														
1	2	13	15																														
9	12	8	3																														
1	2	3	4																														
12	13	14	5																														
11		15	6																														
10	9	8	7																														
<p><b>Trò đố 8 ô</b></p> <table border="1" style="margin: auto;"> <tr><td></td><td>2</td><td>8</td></tr> <tr><td>3</td><td>5</td><td>7</td></tr> <tr><td>6</td><td>4</td><td>1</td></tr> </table>		2	8	3	5	7	6	4	1	<table border="1" style="margin: auto;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>8</td><td></td><td>4</td></tr> <tr><td>7</td><td>6</td><td>5</td></tr> </table>	1	2	3	8		4	7	6	5														
	2	8																															
3	5	7																															
6	4	1																															
1	2	3																															
8		4																															
7	6	5																															



## THUẬT TOÁN - THUẬT GIẢI

Xét bài toán 8 puzzle với goal là:

1	2	3
8		4
7	6	5

**Heuristic 1:** Tổng số miếng sai vị trí

**Heuristic 2:** Tổng khoảng cách sai vị trí của từng miếng.

<table border="1" style="margin: auto;"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td>6</td><td>4</td></tr> <tr><td>7</td><td>5</td><td></td></tr> </table>	2	8	3	1	6	4	7	5		<table border="1" style="margin: auto;"> <tr><td>5</td><td></td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	5				6				
2	8	3																	
1	6	4																	
7	5																		
5																			
	6																		
<table border="1" style="margin: auto;"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td></td><td>4</td></tr> <tr><td>7</td><td>6</td><td>5</td></tr> </table>	2	8	3	1		4	7	6	5	<table border="1" style="margin: auto;"> <tr><td>3</td><td></td><td></td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	3				4				
2	8	3																	
1		4																	
7	6	5																	
3																			
	4																		
<table border="1" style="margin: auto;"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td>6</td><td>4</td></tr> <tr><td>7</td><td>5</td><td></td></tr> </table>	2	8	3	1	6	4	7	5		<table border="1" style="margin: auto;"> <tr><td>5</td><td></td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	5				6				
2	8	3																	
1	6	4																	
7	5																		
5																			
	6																		

Việc chọn lựa hàm Heuristic là khó khăn và có ý nghĩa quyết định đối với tốc độ của giải thuật



## THUẬT TOÁN - THUẬT GIẢI

Xét lại hoạt động của giải thuật Best First Search:

- Khi có 2 nút cùng có giá trị kỳ vọng đạt đến mục tiêu bằng nhau thì nút có path từ nút bắt đầu đến nút đó ngắn hơn sẽ được chọn trước như vậy nút này có giá trị Heuristic tốt hơn.
- Hay nói cách khác hàm lượng giá Heuristic cho nút gần start hơn là tốt hơn nếu kỳ vọng đến goal là bằng nhau.
- Vậy chọn nút nào nếu kỳ vọng của 2 nút khác nhau? Nút kỳ vọng tốt hơn nhưng xa start hay nút kỳ vọng xấu hơn nhưng gần root

Hàm lượng giá bao gồm cả 2 và có cấu trúc:

$$F(n) := G(n) + H(n)$$

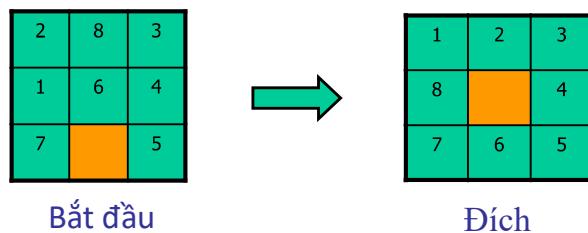
**G(n):** phí tổn thực từ root đến n

**H(n):** phí tổn ước lượng heuristic từ n đến goal.



## THUẬT TOÁN - THUẬT GIẢI

Xét ví dụ là bài toán 8 puzzle với:



Hàm lượng giá:  $F(n) = G(n) + H(n)$

Với  $G(n)$ : số lần chuyển vị trí đã thực hiện

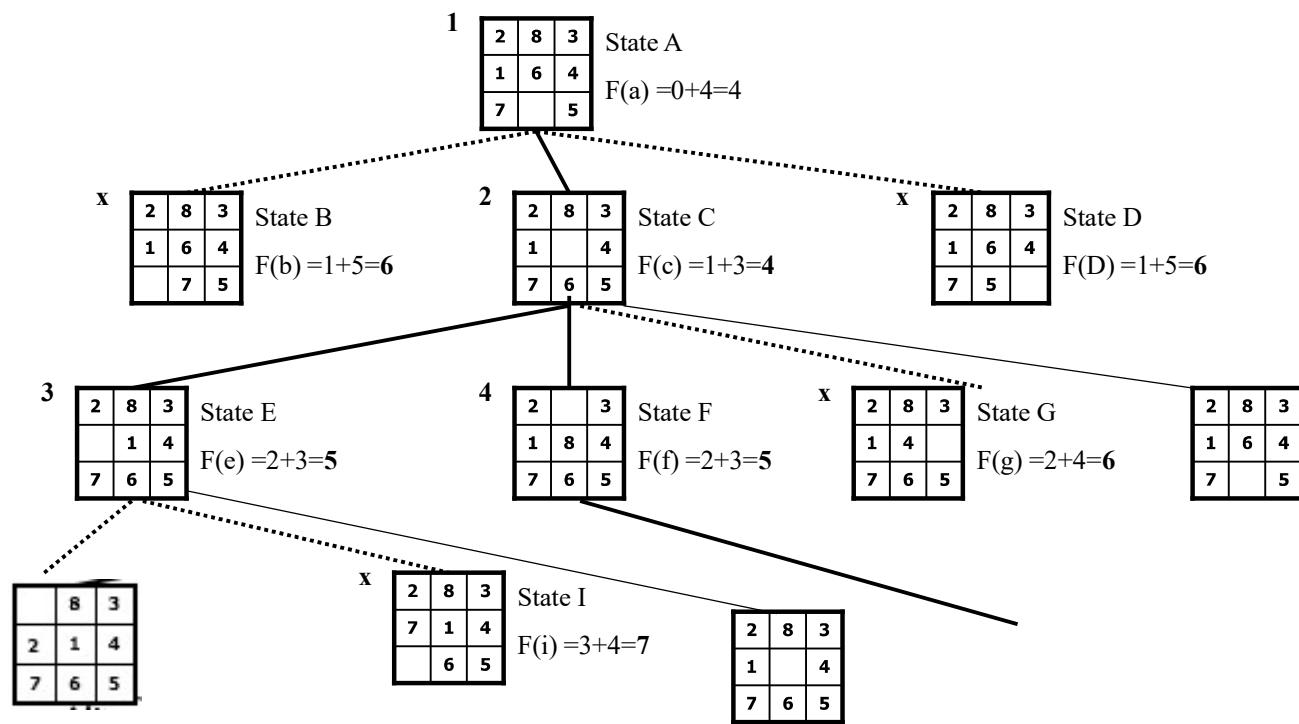
$H(n)$ : Số miếng nằm sai vị trí

Nút X có giá trị heuristic tốt hơn nút Y nếu  $F(x) < F(y)$ .

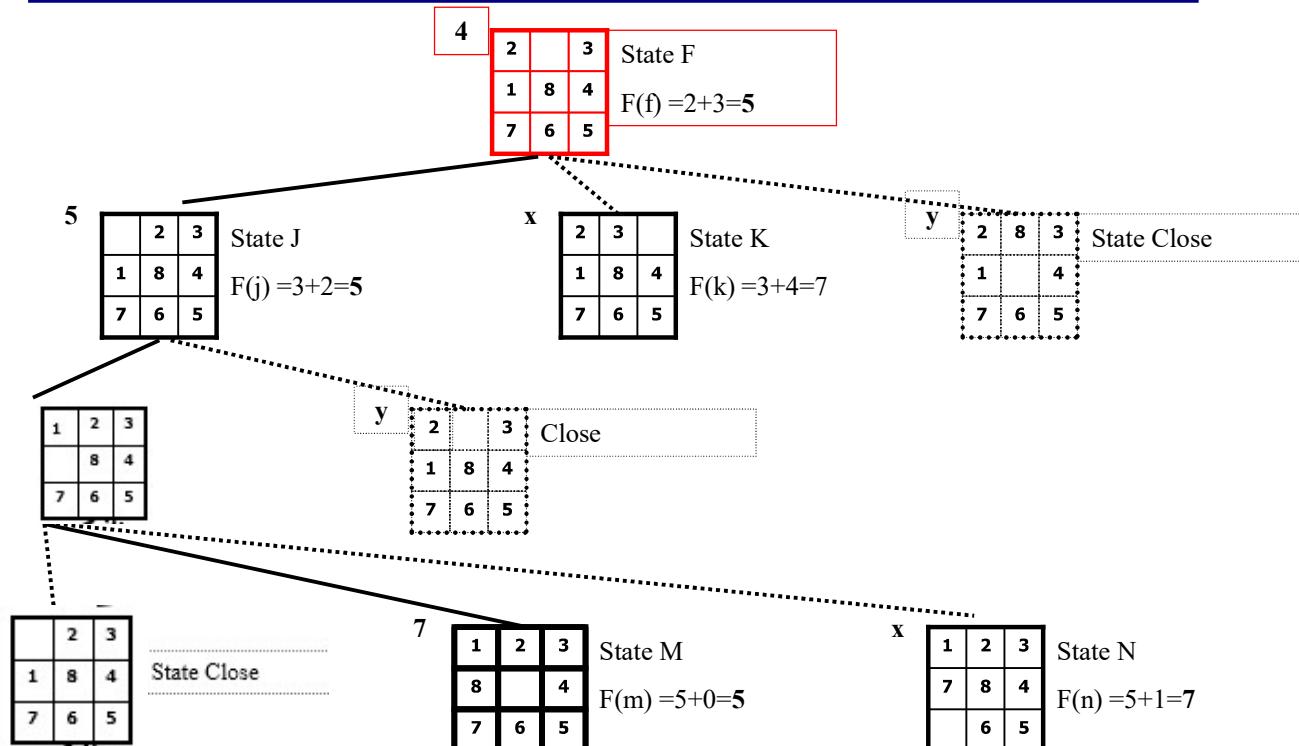
Ta có hoạt động của giải thuật Best First search trên như hình sau:



## THUẬT TOÁN - THUẬT GIẢI



## THUẬT TOÁN - THUẬT GIẢI





## THUẬT TOÁN - THUẬT GIẢI

Lần	n	Open	Close
0		{A4}	{}
1	A4	{C4,B6,D6}	{A4}
2	C4	{E5,F5,G6,B6,D6}	{A4,C4}
3	E5	{F5,H6,G6,B6,D6,I7}	{A4,C4,E5}
4	F5	{J5,H6,G6,B6,D6,K7,I7}	{A4,C4,E5,F5}
5	J5	{L5,H6,G6,B6,D6,K7,I7}	{A4,C4,E5,F5,J5}
6	I5	{M5,H6,G6,B6,D6,K7,I7,N7}	{A4,C4,E5,F5,J5,L5}
7	<b>m5</b>		



## THUẬT TOÁN - THUẬT GIẢI

### Admissibility – Tính chấp nhận

- Một giải thuật Best first search với hàm đánh giá
- $F(n) = G(n) + H(n)$  với
  - ▣ N : Trạng thái bất kỳ
  - ▣ G(n): Phí tổn đi từ nút bắt đầu đến nút n
  - ▣ H(n) : Phí tổn ước lượng heuristic đi từ nút n đến goal
- Được gọi là giải thuật A

Một giải thuật tìm kiếm được xem là admissible nếu đối với một đồ thị bất kỳ nó luôn dừng ở path nghiệm tốt nhất (nếu có).

**Giải thuật A\***: Là giải thuật A với hàm heuristic  $H(n)$  luôn  $\leq$  giá trị thực đi từ n đến goal.

### Giải thuật A\* là admissible



# THUẬT TOÁN - THUẬT GIẢI

## HEURISTIC TRONG TRÒ CHƠI ĐỐI KHÁNG

### Giải thuật minimax:

- Hai đấu thủ trong trò chơi được gọi là **MIN** và **MAX**.
- Mỗi nút lá có giá trị:
  - 1 nếu là MAX thắng.
  - 0 nếu là MIN thắng.

**Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị, qua các nút cha mẹ kế tiếp theo các luật sau:**

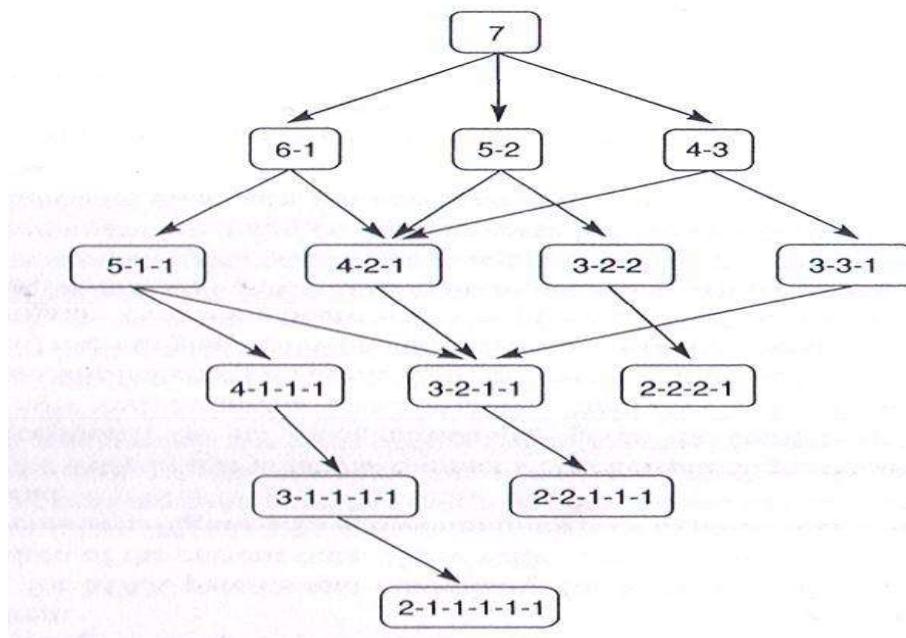
Nếu trạng thái cha mẹ là **MAX**, gán cho nó giá trị **lớn nhất** có trong các trạng thái con.

Nếu trạng thái bố, mẹ là **MIN**, gán cho nó giá trị **nhỏ nhất** có trong các trạng thái con.



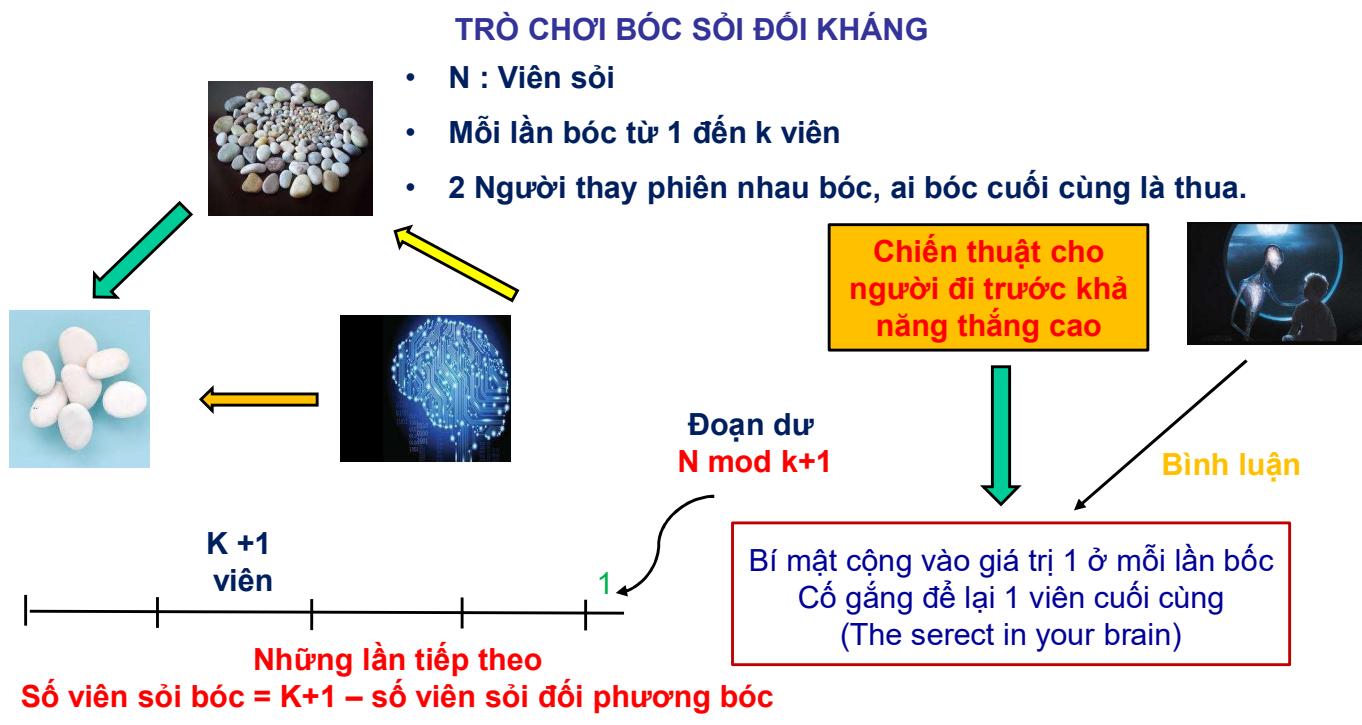
# THUẬT TOÁN - THUẬT GIẢI

## ÁP DỤNG MIN – MAX VÀO TRÒ CHƠI NIM





## THUẬT TOÁN - THUẬT GIẢI



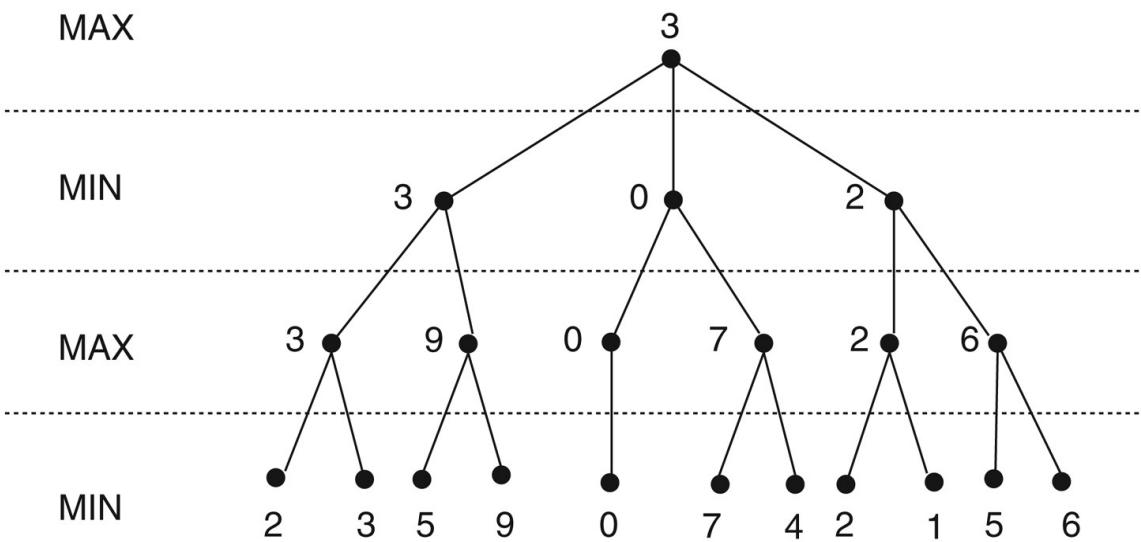
Artificial Intelligence

Nguyễn Đình Cường 43



## THUẬT TOÁN - THUẬT GIẢI

### Minimax với độ sâu lớp cỗ định



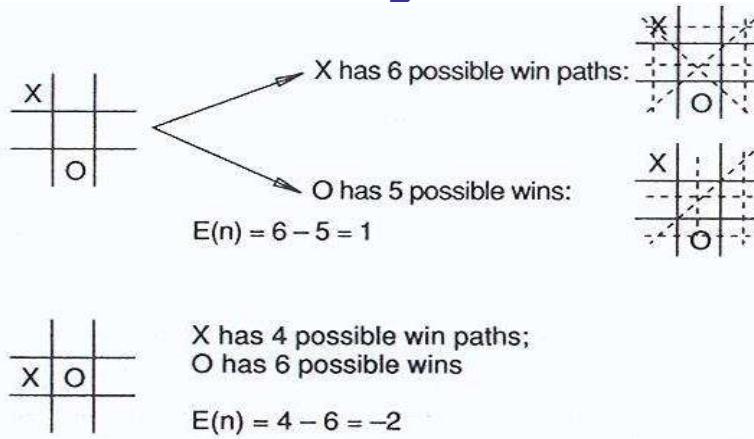
Artificial Intelligence

Nguyễn Đình Cường 44



## THUẬT TOÁN - THUẬT GIẢI

### Heuristic trong trò chơi tic-tac-toe



Hàm Heuristic:  $E(n) = M(n) - O(n)$

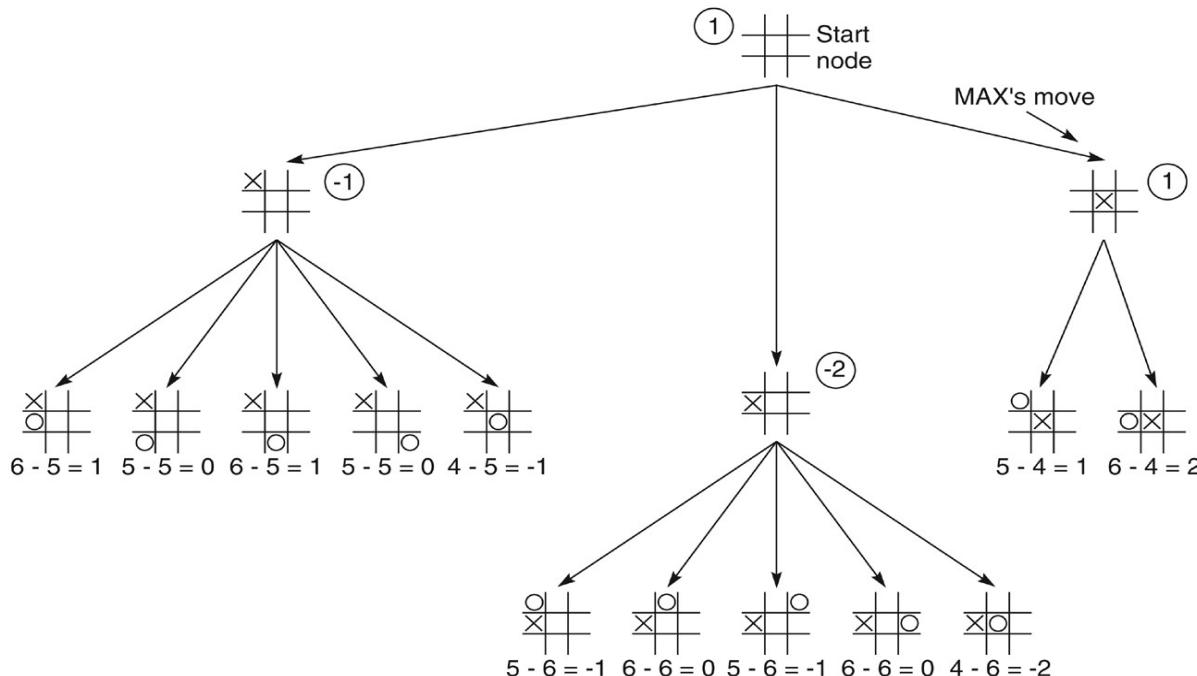
Trong đó:  $M(n)$  là tổng số đường thắng có thể của tôi

$O(n)$  là tổng số đường thắng có thể của đối thủ

$E(n)$  là trị số đánh giá tổng cộng cho trạng thái  $n$



## THUẬT TOÁN - THUẬT GIẢI





## THUẬT TOÁN - THUẬT GIẢI

### GiẢI THUẬT CẮT TỈA $\alpha$ - $\beta$

Tìm kiếm theo kiểu depth-first.

Nút MAX có 1 giá trị  $\alpha$  (luôn tăng)

Nút MIN có 1 giá trị  $\beta$  (luôn giảm)

TK có thể kết thúc dưới bất kỳ:

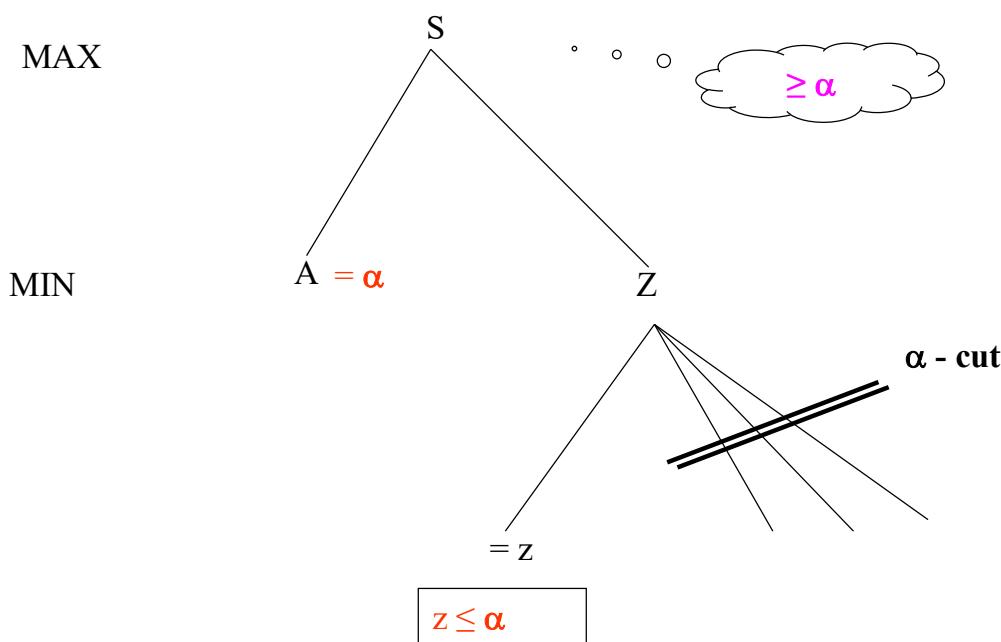
- Nút MIN nào có  $\beta \leq \alpha$  của bất kỳ nút cha MAX nào.
- Nút MAX nào có  $\alpha \geq \beta$  của bất kỳ nút cha MIN nào.

Giải thuật cắt tỉa  $\alpha$ - $\beta$  thể hiện mối quan hệ giữa các nút ở lớp  $n$  và  $n+2$ , mà tại đó toàn bộ cây có gốc tại lớp  $n+1$  có thể cắt bỏ.



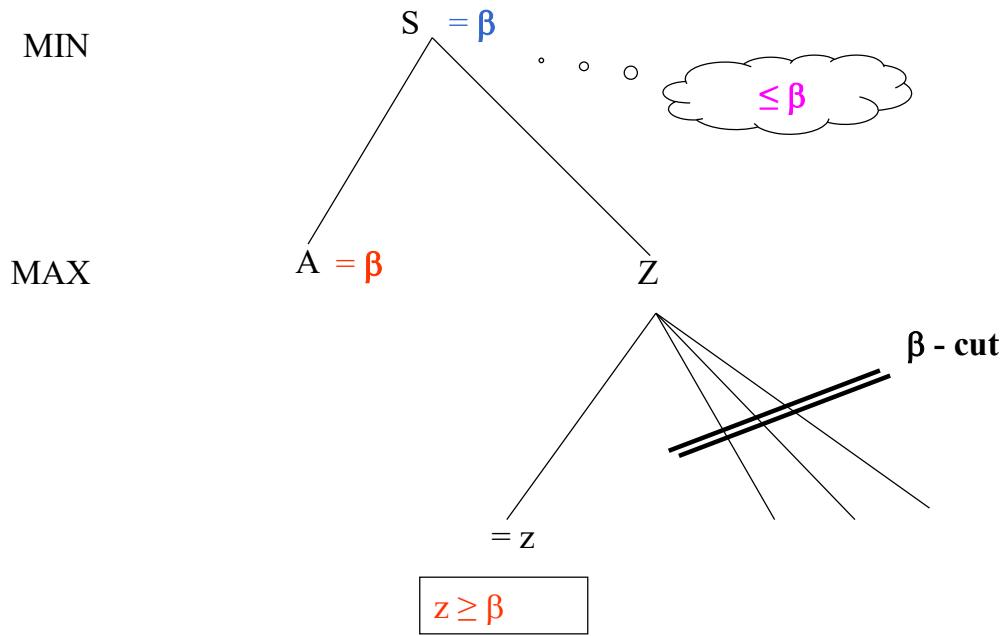
## THUẬT TOÁN - THUẬT GIẢI

### Cắt tỉa $\alpha$

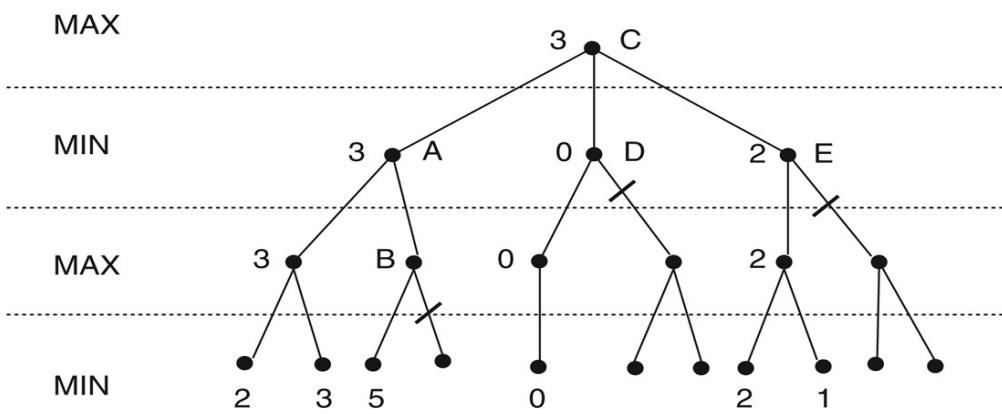




## THUẬT TOÁN - THUẬT GIẢI



## THUẬT TOÁN - THUẬT GIẢI



A has  $\beta = 3$  (A will be no larger than 3)

B is  $\beta$  pruned, since  $5 > 3$

C has  $\alpha = 3$  (C will be no smaller than 3)

D is  $\alpha$  pruned, since  $0 < 3$

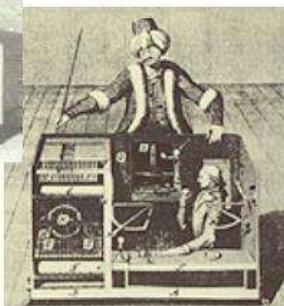
E is  $\alpha$  pruned, since  $2 < 3$

C is 3



Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

Bộ môn Công Nghệ Phần Mềm  
Khoa Công Nghệ Thông Tin

Nha trang - 2012

Artificial Intelligence

Nguyễn Đình Cường 51



## GIẢI THUẬT DI TRUYỀN

### TỔNG QUAN VỀ GA

- **Bài toán tối ưu:** Tìm lời giải tốt nhất trong không gian tìm kiếm.  
Nếu không gian tìm kiếm nhỏ: Áp dụng các phương pháp cổ điển.  
phải dùng đến những kỹ thuật khác: Trí tuệ nhân tạo, GA là thuật giải mô phỏng các hiện tượng tự nhiên: kẽ thừa và đấu tranh sinh tồn để khảo sát không gian lời giải  
Quan niệm: quá trình tiến hóa tự nhiên là quá trình thế hệ sau bao giờ cũng tốt hơn thế hệ trước.
- Tiến hóa tự nhiên được duy trì nhờ 2 quá trình cơ bản: sinh sản và chọn lọc tự nhiên.
- Sự thay đổi môi trường là động lực thúc đẩy quá trình tiến hóa.
- Các cá thể sinh ra trong quá trình tiến hóa nhờ sự lai ghép ở thế hệ bố mẹ. Một cá thể mới có thể mang những đặc tính của thế hệ bố mẹ (di truyền) cũng có thể mang những đặc tính hoàn toàn mới (đột biến). Đột biến xảy ra với xác suất nhỏ hơn nhiều so với di truyền.
- Các thuật toán di truyền dựa trên các quá trình: *lai ghép, đột biến, sinh sản và chọn lọc tự nhiên.*

Artificial Intelligence

Nguyễn Đình Cường 52



## GIẢI THUẬT DI TRUYỀN

### QUÁ TRÌNH LAI GHÉP

Phép lai ghép với xác suất  $p_c$ , được mô phỏng như sau:

- Chọn ngẫu nhiên hai (hay nhiều) cá thể bất kỳ trong quần thể. Giả sử rằng, các nhiễm sắc thể của thế hệ bố mẹ đều có m gen.
- Tạo ngẫu nhiên một số  $i$ :  $1 \leq i \leq m-1$  ( $i$  được gọi là điểm lai). Điểm lai chia chuỗi bối mẹ thành 2 chuỗi con dài  $m_{11} m_{12}$  và  $m_{21} m_{22}$ . Hai chuỗi nhiễm sắc thể con mới là  $m_{11}m_{22}$  và  $m_{21}m_{12}$ .
- Đưa 2 cá thể mới này vào quần thể để tham gia các quá trình tiến hóa tiếp theo.



## GIẢI THUẬT DI TRUYỀN

### QUÁ TRÌNH ĐỘT BIẾN

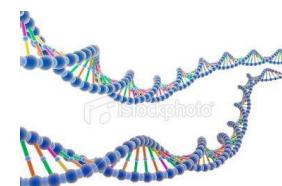
- Cá thể con mang một số tính trạng không có trong mã di truyền của bố mẹ.
- Phép đột biến xảy ra với xác suất  $p_m$ , nhỏ hơn rất nhiều so với xác suất lai ghép  $p_c$ .

Phép đột biến được mô tả:

Chọn ngẫu nhiên 1 cá thể bất kỳ trong quần thể.

Tạo một số ngẫu nhiên k:  $1 \leq k \leq m$

Thay đổi gen thứ k và đưa cá thể này vào quần thể để tham gia quá trình tiến hóa tiếp theo.





## GIẢI THUẬT DI TRUYỀN

### QUÁ TRÌNH SINH SẢN

- Phép tái sinh là quá trình các cá thể được sao chép trên cơ sở độ thích nghi của nó. Độ thích nghi là một hàm gán một giá trị thực cho các cá thể trong quần thể.
- Quá trình sinh sản được mô phỏng:
  - ▣ Tính độ thích nghi cho từng cá thể trong quần thể hiện hành  $F_i$  lập bảng cộng dồn các độ thích nghi  $F_{ti}$  và tính tổng độ thích nghi của toàn quần thể  $F_m$ .
  - ▣ Tạo số ngẫu nhiên  $F$ :  $0 \leq F \leq F_m$ .
  - ▣ Chọn cá thể thứ  $k$  đầu tiên thỏa  $F \geq F_{tk}$  và đưa vào quần thể của thế hệ mới.



Artificial Intelligence

Nguyễn Đình Cường 55



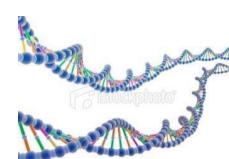
## GIẢI THUẬT DI TRUYỀN

### QUÁ TRÌNH CHỌN LỌC

Giữ lại các cá thể tốt và loại bỏ các cá thể xấu trong quần thể.

Phép chọn được mô phỏng như sau:

- Sắp xếp quần thể theo thứ tự độ thích nghi giảm dần.
- Loại bỏ các cá thể cuối dãy, chỉ giữ lại  $n$  cá thể tốt nhất. Giả sử  $n$  là kích thước của quần thể (pop\_size)



Artificial Intelligence

Nguyễn Đình Cường 56



## GIẢI THUẬT DI TRUYỀN

GA là thuật giải mô phỏng các hiện tượng tự nhiên: kế thừa và đấu tranh sinh tồn để tìm kiếm lời giải tối ưu.

*Ví dụ về quần thể thỏ:* Một số con nhanh nhẹn hơn những con khác có xác suất bị chồn cáo ăn thịt nhỏ hơn, chúng tồn tại và tạo thêm nhiều thỏ tốt, dĩ nhiên cũng có một số con thỏ chậm chạp sống chỉ vì may mắn. Quần thể những con thỏ sống sót sẽ sinh sản ra các con thỏ với các đặc điểm từ các thỏ bố mẹ và thỉnh thoảng lại có một con thỏ đột biến nào đó.

**Để tìm kiếm lời giải tối ưu, GA cũng thực hiện các bước tương ứng như câu chuyện đấu tranh sinh tồn của loài thỏ.**

**GA sử dụng thuật ngữ được vay mượn từ Di truyền học:**

Cá thể (hay kiểu gen, cấu trúc) trong quần thể: các cá thể còn được gọi là các chuỗi hay các nhiễm sắc thể.

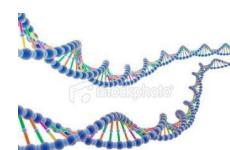


## GIẢI THUẬT DI TRUYỀN

### NĂM THÀNH PHẦN TRONG VIỆC ỨNG DỤNG THUẬT GIẢI DI TRUYỀN

Cấu trúc dữ liệu biểu diễn không gian lời giải của bài toán.

- Phương pháp khởi tạo quần thể ban đầu P(0).
- Hàm xác định độ thích nghi Eval( ), đóng vai trò môi trường
- Các phép toán di truyền (lai ghép, đột biến, sinh sản và chọn lọc tự nhiên)
- Các tham số cần sử dụng trong thuật giải (kích thước quần thể, xác suất lai, xác suất đột biến, ...)





## GIẢI THUẬT DI TRUYỀN

### Genetic Algorithm

Begin

t=0;

Khởi tạo P(t);

Tính độ thích nghi của các cá thể thuộc P(t);

While (điều kiện dừng chưa thỏa) do

Begin

t:=t+1;

Chọn lọc P(t) từ P(t-1);

Kết hợp các cá thể của P(t) ;

Tính độ thích nghi của các cá thể thuộc P(t);

End;

End

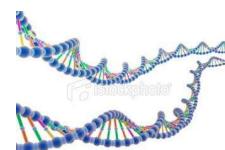
Ghi chú: Để giải được bài toán theo GA cần thiết phải vận dụng tri thức bài toán sao Cho xây dựng được cấu trúc dữ liệu hợp lý để áp dụng được giải thuật di truyền một cách tự nhiên và hiệu quả.



## GIẢI THUẬT DI TRUYỀN

### BÀI TOÁN TSP(TRAVELING SALES PROBLEM)

- Một người bán hàng cần đến tất cả các thành phố, mỗi thành phố một lần rồi trở về điểm khởi hành, biết trước quãng đường (chi phí) giữa 2 thành phố bất kỳ. Hãy tìm lộ trình đi sao cho tổng quãng đường (hay tổng chi phí là nhỏ nhất).
- Bài toán TSP có thể tiếp cận: nhánh cận, phương pháp gần đúng hay Heuristic. Sau đây, xét bằng giải thuật GA:

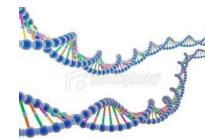




# GIẢI THUẬT DI TRUYỀN

## 1. Biểu diễn nhiễm sắc thể

- Cần phải tìm cách biểu diễn nhiễm sắc thể cho phù hợp với bài toán.
- Nếu dùng chuỗi nhị phân biểu diễn cho Bài toán TSP có  $n$  thành phố, mỗi thành phố phải được mã bằng chuỗi  $\log_2(n)$  bit và nhiễm sắc thể là chuỗi gồm  $n \cdot \log_2(n)$  bit.
- Tuy nhiên, phép lai tạo, đột biến có thể tạo ra một lộ trình không thỏa mãn bài toán, chẳng hạn, một thành phố đến 2 lần.
- Ngoài ra, trong trường hợp  $n$  không là lũy thừa của 2 thì sẽ có một số dãy bit không tương ứng với thành phố nào cả. Ví dụ,  $n=20$ , khi đó cần dùng 5 bit để biểu diễn 1 thành phố và dãy 10101 không tương ứng với 1 thành phố nào cả.
- Như vậy, ta có thể sử dụng một cách tự nhiên là đánh số các thành phố và dùng vectơ gồm các số nguyên để biểu diễn một nhiễm sắc thể lộ trình.



# GIẢI THUẬT DI TRUYỀN

## 2. Khởi tạo quần thể ban đầu: Có thể tạo bằng một trong các cách sau:

Tạo ngẫu nhiên các mẫu từ các hoán vị của  $1, 2, \dots, n$

Áp dụng các Heuristic, chẳng hạn, dùng Heuristic Greedy nhiều lần, mỗi lần từ một thành phố đầu khác nhau.

## 3. Hàm lượng giá: Cho trước lộ trình ta có thể tính tổng khoảng cách (chi phí) của lộ trình đó.

## 4. Phép toán di truyền:

Phép lai ghép: Cần xây dựng phép lai ghép sao cho cá thể con phải bảo toàn thứ tự giữa các thành phố. Ví dụ cho cá thể bố mẹ là:

<1 2 3 4 5 6 7 8 9 10 11 12>

<7 3 1 11 4 12 5 2 10 9 6 8>

Giả sử khúc được chọn là 4 5 6 7, cá thể con của phép lai ghép sẽ là:

<1 11 12 4 5 6 7 2 10 9 8 3>

Có thể hoán đổi vai trò bố mẹ để xây dựng cá thể con thứ 2

Phép đột biến: chỉ cần hoán vị 2 vị trí bất kỳ trong cá thể được chọn

Ví dụ: với nhiễm sắc thể <7 3 1 11 4 12 5 2 10 9 6 8> đột biến tại vị trí số 3 và vị trí 5, ta có

<7 3 4 11 1 12 5 2 10 9 6 8>

Ghi chú: Đây chỉ là một cách định nghĩa các phép toán di truyền. Với các Heuristic khác có thể thực hiện phép toán tốt hơn

## 5. Các tham số: như trên.



# GIẢI THUẬT DI TRUYỀN

## CƠ CHẾ THỰC HIỆN CỦA GA

### Mở đầu

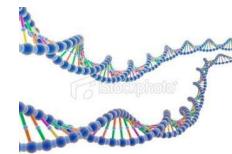
Không mất tính tổng quát, giả sử bài toán tối ưu là bài toán tìm giá trị cực đại và hàm mục tiêu  $f$  có giá trị dương trên miền xác định của nó.

Bởi vì bài toán tìm cực tiểu hàm  $f$  chính là tìm cực đại hàm  $g = -f$   
 $\min\{f(x)\} = \max\{g(x)\} = \max\{-f(x)\}$

Và nếu hàm mục tiêu  $f$  không dương trên miền xác định của nó, ta cộng thêm hằng số dương  $C$  đủ lớn sao cho:

$$\max\{g(x)\} = \max\{f(x)+C\}$$

Khi đó hai vế trái và phải cùng đạt cực đại tại điểm  $x_0$ .



Artificial Intelligence

Nguyễn Đình Cường 63



# GIẢI THUẬT DI TRUYỀN

Bài toán: Tìm giá trị lớn nhất của hàm k biến

$$\begin{array}{ccc} f: & R_k & \longrightarrow R \\ & (x_1, x_2, \dots, x_k) & \longrightarrow f(x_1, x_2, \dots, x_k) \end{array}$$

1. Biểu diễn nhiễm sắc thể:

Giả sử mỗi biến  $x_i$  nhận giá trị trong miền  $D_i = [a_i, b_i]$  với  $i=1,..,k$

và  $f(x_1, x_2, \dots, x_k) > 0 \forall x_i \in D_i$ . Bài toán đặt ra cần tối ưu hóa hàm  $f$  với độ chính xác cho trước, chẳng hạn 6 số lẻ đổi với giá trị các biến.

Để thỏa độ chính xác trên, mỗi miền  $D_i$  được chia thành  $(b_i - a_i)*106$  miền con bằng nhau. Gọi  $m_i$  là số nguyên dương nhỏ nhất sao cho:

$$(b_i - a_i)*106 \leq 2m_i - 1$$

Như vậy, mỗi biến  $x_i$  được biểu diễn thành chuỗi có độ dài  $m_i$  bit. Với mỗi chuỗi nhị phân  $m_i$  bit đổi sang số thực tương ứng trong  $[a_i, b_i]$  như sau :

Đổi chuỗi nhị phân sang số trong hệ thập phân  $t_i$

Tính Khi đó, mỗi nhiễm sắc thể (là một lời giải) được biểu diễn bằng chuỗi nhị phân có chiều dài trong đó  $m_1$  bit đầu biểu diễn cho giá trị trong khoảng  $[a_1, b_1]$ ,  $m_2$  bit tiếp theo biểu diễn cho giá trị trong khoảng  $[a_2, b_2]$ , ..,  $m_k$  bit cuối biểu diễn các giá trị trong khoảng  $[a_k, b_k]$

Artificial Intelligence

Nguyễn Đình Cường 64



## GIẢI THUẬT DI TRUYỀN

### 2. Khởi tạo quần thể

Với kích thước quần thể cho trước, ta tạo các cá thể với các nhiễm sắc thể ngẫu nhiên theo từng bit. Có thể sử dụng phân phối xác suất để khởi tạo quần thể ban đầu.

### 3. Hàm lượng giá

Lượng giá từng nhiễm sắc thể bằng cách tính giá trị hàm  $f$  trên các chuỗi nhị phân đã được giải mã.

### 4. Phép toán di truyền

Thực hiện phép lai ghép và đột biến để tạo các cá thể mới dựa trên độ thích nghi. Sau một số thế hệ, khi không còn cải thiện thêm được nữa, ta cho thuật toán dừng và nhiễm sắc thể tốt nhất được xem là lời giải tối ưu.



## GIẢI THUẬT DI TRUYỀN

### 2.2 Xác định độ thích nghi bằng bánh xe Rulet

- Trong phần này giả sử các độ thích nghi đều dương, trong trường hợp ngược lại, sử dụng các phép biến đổi tương ứng để định lại tỷ lệ sao cho độ thích nghi đều dương
- Tính độ thích nghi  $Eval(v_i)$  cho mỗi nhiễm sắc thể  $v_i$  ( $i=1,..,pop\_size$ )
- Tính tổng giá trị thích nghi của toàn quần thể
- Tính xác suất chọn  $p_i$  cho mỗi nhiễm sắc thể  $v_i$  :
$$p_i = Eval(v_i)/F$$
- Tính vị trí xác suất  $q_i$  của mỗi nhiễm sắc thể  $v_i$
- Quá trình chọn lọc bằng cách quay bánh xe Rulet  $pop\_size$  lần, mỗi lần chọn một nhiễm sắc thể từ quần thể hiện hành vào quần thể mới như sau :
- Tạo số ngẫu nhiên  $r \in [0,1]$
- Nếu  $r < q_1$  thì chọn nhiễm sắc thể đầu tiên  $v_1$ , ngược lại chọn nhiễm sắc thể  $v_i$  với  $q_{i-1} \leq r \leq q_i$
- Ghi chú : Có thể xảy ra trường hợp một số nhiễm sắc thể được chọn nhiều lần. Điều này phù hợp với Lý thuyết sơ đồ: các nhiễm sắc thể tốt thì có nhiều bản sao hơn.



## GIẢI THUẬT DI TRUYỀN

Các phép toán di truyền

### Phép lai tạo

- Cho xác suất lai là  $p_c$ , số nhiễm sắc thể được dùng để lai tạo là  $\text{pop\_size} * p_c$ . Việc lai tạo tiến hành như sau :
- Đối với mỗi nhiễm sắc thể trong quần thể mới :
- Tạo số ngẫu nhiên  $r \in [0,1]$

### Nếu $r < p_c$ chọn

Lấy mỗi cặp nhiễm sắc thể lai tạo ra cặp cá thể con như sau: tạo số nguyên ngẫu nhiên  $s \in [1, m-1]$  với  $m$  là số bít của 1 nhiễm sắc thể.

- Hai nhiễm sắc thể bố mẹ :  
 $(b_1 \ b_2 \dots \ b_s \ b_{s+1} \ \dots \ b_m)$  và  $(c_1 \ c_2 \dots \ c_s \ c_{s+1} \ \dots \ c_m)$
- Được thay bằng cặp con của chúng  
 $(b_1 \ b_2 \dots \ b_s \ c_{s+1} \ \dots \ c_m)$  và  $(c_1 \ c_2 \dots \ c_s \ b_{s+1} \ \dots \ b_m)$

### Phép đột biến

- Cho xác suất đột biến là  $p_m$ , số bit đột biến sẽ là  $p_m * m * \text{pop\_size}$ . Mỗi bit trong tất cả các nhiễm sắc thể có cơ hội đột biến như nhau, nghĩa là, đổi từ 0 sang 1 hoặc ngược lại.
- Với mỗi nhiễm sắc thể trong quần thể hiện hành (sau khi lai) và đổi với mỗi bit trong nhiễm sắc thể:
  - Tạo số ngẫu nhiên  $r \in [0,1]$ . Nếu  $r < p_m$  thì đột biến bit này.
  - Sau quá trình chọn lọc, lai và đột biến, quần thể mới được lượng giá kế tiếp.



## GIẢI THUẬT DI TRUYỀN

### GIẢI BÀI TOÁN TỐI ƯU SỐ

Cần tìm giá trị lớn nhất hàm sau:

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2) \text{ với } -3.0 \leq x_1 \leq 12.1 \text{ và } 4.1 \leq x_2 \leq 5.8$$

Cho trước kích thước quần thể  $\text{pop\_size}=20$ ,  $p_c=0.25$  và  $p_m=0.01$

Giả sử cần tính chính xác đến 4 số lẻ đối với mỗi biến.

Biểu diễn nhiễm sắc thể cho các biến:

Miền của biến  $x_1$  được chia thành  $15.1 * 10^4 = 151000$  khoảng cần 18 bit vì  $2^{17} \leq 151000 \leq 2^{18}$

Miền của biến  $x_2$  được chia thành  $1.7 * 10^4 = 17000$  khoảng cần 15 bit vì  $2^{14} \leq 17000 \leq 2^{15}$

Chiều dài của nhiễm sắc thể (vectơ lời giải) là 33 bit, trong đó 18 bit đầu mã hóa  $x_1$ , 15 bit sau mã hóa  $x_2$

Ví dụ: Xét 1 nhiễm sắc thể:  $v = (010001001011010000111110010100010)$

Trong đó 18 bit đầu biểu diễn số:  $x_1 = -3.0 + \text{decimal}(010001001011010000) * = -3.0 + 4.052426$

Tương tự, 15 bit sau  $(111110010100010)$  biểu diễn số:  $x_2 = 4.1 * \text{decimal}(111110010100010) * = 4.1 + 1.1655330$

Như vậy, nhiễm sắc thể  $v = (010001001011010000111110010100010)$

tương ứng với  $\langle x_1, x_2 \rangle = \langle 1.052426, 5.755330 \rangle$

Độ thích nghi của nhiễm sắc thể này là  $f(1.052426, 5.755330) = 20.252640$



## GIẢI THUẬT DI TRUYỀN

Khởi tạo quần thể ban đầu: tạo quần thể gồm pop\_size=20 nhiễm sắc thể, tất cả 33 bit của mỗi nhiễm sắc thể được tạo ngẫu nhiên. Giả sử, sau khi khởi tạo có quần thể sau:

```

v1 = 100100100000011111101001101111
v2 = 111000100100110111001010100011010
v3 = 0000100001100100001010111011101
v4 = 100011000101101001111000001110010
v5 = 00011101100101001101011111000101
v6 = 00010100001001010100101000000011
v7 = 00100010000110101111011011111011
v8 = 100001100001110100010110101100111
v9 = 010000000101100010110000001111100
v10= 000001111000110000011010000111011
v11= 01100111110110101100001101111000
v12= 110100010111101101000101010000000
v13= 111011111010001000110000001000110
v14= 010010011000001010100111100101001
v15= 11101110110111000100011111011110
v16= 11001111000001111100001101001011
v17= 01101011111001111010001101111101
v18= 01110100000001110100111110101101
v19= 0001010100111111110000110001100
v20= 10111001011001111001100010111110

```



## GIẢI THUẬT DI TRUYỀN

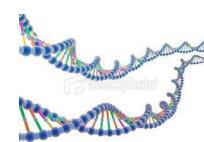
Xác định độ thích nghi:

Eval(v1) = f(6.084492,5.652242) =	26.019600
Eval(v2) = f(10.348434,4.380264) =	7.580015
Eval(v3) = f(-2.516603,4.390381) =	19.626329
Eval(v4) = f(5.278638,5.593460) =	17.406725
Eval(v5) = f(-1.255173,4.734458) =	25.341160
Eval(v6) = f(-1.811725, 4.391937) =	18.100417
Eval(v7) = f(-0.991471, 5.680258) =	16.020812
Eval(v8) = f(4.910618,4.703018) =	17.959701
Eval(v9) = f(0.795406,5.381472) =	16.127799
Eval(v10) = f(-2.554851,4.793707) =	21.278435
Eval(v11) = f(3.130078,4.996097) =	23.410669
Eval(v12) = f(9.356179,4.239457) =	15.011619
Eval(v13) = f(11.134646,5.378671) =	27.316720
Eval(v14) = f(1.335944,5,151378) =	19.876294
Eval(v15) = f(11.089025,5.054515) =	30.060205
Eval(v16) = f(9.211598,4.993762) =	23.967227
Eval(v17) = f(3.367514,4.571343) =	13.696165
Eval(v18) = f(3.843020,5,158226) =	15.414128
Eval(v19) = f(-1.746635,5.395584) =	20.095903
Eval(v20) = f(7.935998,4.757338) =	13.666916

Ta có nhiễm sắc thể v15 mạn nhất và v2 yếu nhất.

$$F = \sum Eval(v_i) = 387.976821$$

Xác suất chọn  $p_i = Eval(v_i)/F$  và vị trí xác suất  
Ta có bảng trang 11 và 12



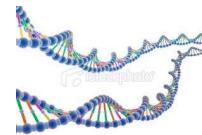


## GIẢI THUẬT DI TRUYỀN

Cho quay bánh xe Rulet 20 lần, mỗi lần chọn nhiễm sắc thể cho quần thể mới. Giả sử 20 số ngẫu nhiên trong  $[0,1]$  được tạo ra là :

0.513870	0.175741	0.308652	0.534534	0.947628
0.171736	0.702231	0.226431	0.484773	0.424720
0.703899	0.389647	0.277226	0.368071	0.983437
0.005389	0.7656882	0.6464773	0.767139	0.780237

Số đầu tiên cho ta chọn v11, số thứ 2 chọn v4, ...



## GIẢI THUẬT DI TRUYỀN

Ta có dãy các nhiễm sắc thể được chọn và quần thể mới gồm các nhiễm sắc thể sau:

v'1 = 01100111110110101100001101111000 (v11)  
v'2 = 100011000101101001111000001110010 (v4)  
v'3 = 00100010000011010111101101111011 (v7)  
v'4 = 01100111110110101100001101111000 (v11)  
v'5 = 0001010100111111110000110001100 (v19)  
v'6 = 100011000101101001111000001110010 (v4)  
v'7 = 111011101101110000100011111011110 (v15)  
v'8 = 00011101100101001101011111000101 (v5)  
v'9 = 01100111110110101100001101111000 (v11)  
v'10= 000010000011001000001010111011101 (v3)  
v'11 = 111011101101110000100011111011110 (v15)  
v'12= 01000000010110001011000001111100 (v9)  
v'13= 000101000010010101001010000000011 (v6)  
v'14= 100001100001110100010110101100111 (v8)  
v'15= 101110010110011110011000101111110 (v20)  
v'16= 10010010000001111110100110111111 (v1)  
v'17= 0000011110001100000110100001110111 (v10)  
v'18= 11101111101000100011000001000110 (v13)  
v'19= 111011101101110000100011111011110 (v15)  
v'20= 110011110000011111100001101001011 (v16)



## GIẢI THUẬT DI TRUYỀN

- Áp dụng phép lai tạo

Lai các cá thể trong quần thể mới: Áp dụng phép toán kết hợp, lai các cá thể trong quần thể mới (các vectơ  $v_i$ ). Với xác suất lai  $p_c=0.25$ , đổi với mỗi nhiễm sắc thể ta tạo ra một số ngẫu nhiên  $r \in [0,1]$ . Nếu  $r < 0.25$  ta chọn nhiễm sắc thể đó để lai tạo. Giả sử thứ tự các số ngẫu nhiên  
Như vậy, các nhiễm sắc thể  $v'2, v'11, v'13, v'18$  được chọn để lai tạo.

Ứng với mỗi cặp, ta tạo 1 số ngẫu nhiên là vị trí ghép rồi tạo ra cặp nhiễm sắc thể con từ mỗi cặp.

- Phép đột biến

Với xác suất  $p_m=0.01$ , được thực hiện trên từng bit một. Có  $m * pop\_size = 33 * 20 = 660$  bit trong toàn quần thể. Các bit có cơ hội đột biến như nhau. Với mỗi bit trong quần thể, ta tạo ra một số ngẫu nhiên  $r \in [0,1]$ , nếu  $r < 0.01$  thì bit đó được đột biến.

Chẳng hạn, tại các vị trí bit sau đây có  $r < 0.01$

- Tính độ thích nghi của từng cá thể.

Như vậy, ta đã vừa hoàn thành một bước lặp, thông thường thế hệ sau sẽ tốt hơn thế hệ trước.

Toàn bộ tiến trình trên được lặp lại nhiều thế hệ: hàng trăm, hàng ngàn thế hệ ta sẽ có lời giải tối ưu.

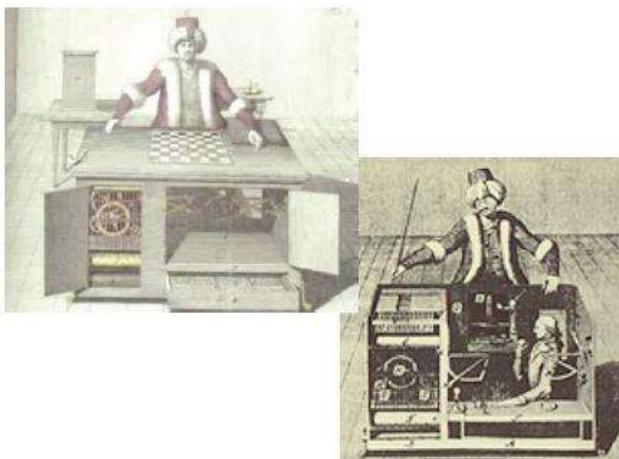
Theo dõi cá thể tốt là bước cần thực hiện trong khi cài đặt giải thuật di truyền. Cá thể trội "hơn" cả sẽ được lưu trữ tại một vị trí riêng biệt. Bằng cách này, thuật giải có thể duy trì cá thể tốt nhất tìm được trong suốt quá trình.



## TRƯỜNG ĐẠI HỌC NHA TRANG

### Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

Bộ môn Công Nghệ Phần Mềm

Khoa Công Nghệ Thông Tin



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Mở đầu

*Tri thức, lĩnh vực và biểu diễn tri thức.*

Các loại tri thức: được chia thành 5 loại

- Tri thức thủ tục: mô tả cách thức giải quyết một vấn đề.  
Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó.  
Các dạng tri thức thủ tục tiêu biểu thường là các luật, chiến lược, lịch trình và thủ tục.
- Tri thức khai báo: cho biết một vấn đề được thấy như thế nào.  
Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng các khẳng định logic đúng hoặc sai.  
Tri thức khai báo cũng có thể là một danh sách các khẳng định nhằm mô tả đầy đủ hơn về đối tượng hay một khái niệm nào đó.

---

Artificial Intelligence

Nguyễn Đình Cường 75



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Siêu tri thức: mô tả *tri thức về tri thức*. Loại tri thức này giúp lựa chọn tri thức thích hợp nhất trong số các tri thức khi giải quyết một vấn đề. Các chuyên gia sử dụng tri thức này để điều chỉnh hiệu quả giải quyết vấn đề bằng cách hướng các lập luận về miền tri thức có khả năng hơn cả.

Tri thức heuristic: mô tả các "*kinh nghiệm*" để dẫn dắt tiến trình lập luận. Tri thức heuristic là *tri thức không bảm đảm hoàn toàn 100% chính xác* về kết quả giải quyết vấn đề. Các chuyên gia thường dùng các tri thức khoa học như sự kiện, luật, ... sau đó chuyển chúng thành các tri thức heuristic để thuận tiện hơn trong việc giải quyết một số bài toán.

Tri thức có cấu trúc: mô tả tri thức theo cấu trúc. Loại tri thức này mô hình tổng quan hệ thống theo quan điểm của chuyên gia, bao gồm khái niệm, khái niệm con, và các đối tượng; diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định.

---

Artificial Intelligence

Nguyễn Đình Cường 76



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### CÁC KỸ THUẬT BIỂU DIỄN TRI THỨC

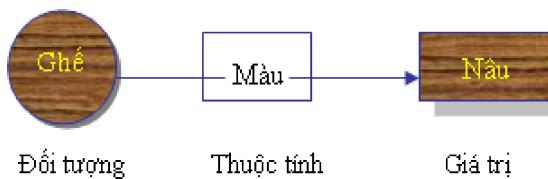
- Bộ ba Đôi tượng-Thuộc tính-Giá trị
- Các luật dẫn
- Mạng ngữ nghĩa
- Frames
- Logic



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### BỘ BA ĐỐI TƯỢNG-THUỘC TÍNH-GIÁ TRỊ

Một sự kiện có thể được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đôi tượng. Ví dụ, mệnh đề "quả bóng màu đỏ" xác nhận "đỏ" là giá trị thuộc tính "màu" của đối tượng "quả bóng". Kiểu sự kiện này được gọi là bộ ba Đôi tượng-Thuộc tính-Giá trị (O-A-V – Object-Attribute-Value).



Hình Biểu diễn tri thức theo bộ ba O-A-V



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

- Trong các sự kiện O-A-V, một đối tượng có thể có nhiều thuộc tính với các kiểu giá trị khác nhau. Hơn nữa một thuộc tính cũng có thể có một hay nhiều giá trị. Chúng được gọi là các sự kiện *đơn trị* (single-valued) hoặc *đa trị* (multi-valued). Điều này cho phép các hệ tri thức linh động trong việc biểu diễn các tri thức cần thiết.
- Các sự kiện không phải lúc nào cũng bảo đảm là đúng hay sai với độ chắc chắn hoàn toàn. Vì thế, khi xem xét các sự kiện, người ta còn sử dụng thêm một khái niệm là *độ tin cậy*. Phương pháp truyền thống để quản lý thông tin không chắc chắn là sử dụng nhân tố chắc chắn CF (certainly factor). Khái niệm này bắt đầu từ hệ thống MYCIN (khoảng năm 1975), dùng để trả lời cho các thông tin suy luận. Khi đó, trong sự kiện O-A-V sẽ có thêm một giá trị xác định độ tin cậy của nó là CF.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### CÁC LUẬT DẪN

- Luật là cấu trúc tri thức dùng để liên kết thông tin đã biết với các thông tin khác giúp đưa ra các suy luận, kết luận từ những thông tin đã biết.
- Trong hệ thống dựa trên các luật, người ta thu thập các tri thức lĩnh vực trong một tập và lưu chúng trong cơ sở tri thức của hệ thống. Hệ thống dùng các luật này cùng với các thông tin trong bộ nhớ để giải bài toán. Việc xử lý các luật trong hệ thống dựa trên các luật được quản lý bằng một module gọi là *bộ suy diễn*.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Các dạng luật cơ bản: 7 dạng

### 1. Quan hệ

IF Bình điện hỏng

THEN Xe sẽ không khởi động được

### 2. Lời khuyên

IF Xe không khởi động được

THEN Đi bộ

### 3. Hướng dẫn

IF Xe không khởi động được AND Hệ thống nhiên liệu tốt

THEN Kiểm tra hệ thống điện



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### 4. Chiến lược

IF Xe không khởi động được

THEN Đầu tiên hãy kiểm tra hệ thống nhiên liệu, sau đó kiểm tra hệ thống điện

### 5. Diễn giải

IF Xe nổ AND tiếng giòn

THEN Động cơ hoạt động bình thường

### 6. Chẩn đoán

IF Sốt cao AND hay ho AND Họng đỏ

THEN Viêm họng

### 7. Thiết kế

IF Là nữ AND Da sáng

THEN Nên chọn Xe Spacy AND Chọn màu sáng

Nếu cố gắng học thì sẽ thi đỗ

Nếu đã cố gắng học thì đã thi đỗ rồi

Nếu thực tập nhiều thì sẽ viết chương trình tốt



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Suy diễn tiến

Từ giả thiết và các luật, suy luận ra thêm những yếu tố mới

	Bước	Kết quả	Luật	Vết
1. $a \rightarrow b$				
2. $a \rightarrow c$	0	{a}		r1   b
3. $bc \rightarrow e$				
4. $e \rightarrow f$	1	{a, b, c}	{r1, r2}	r2   c
Hỏi $a \rightarrow f$	2	{a, b, c, e}	{r1, r2, r3}	r3   e
	3	{a, b, c, e, f}	{r1, r2, r3, r4}	r4   f
		Kết quả:		
		$\begin{array}{l} r1 \\ a \rightarrow b \\ r2 \\ a \rightarrow c \\ bc \rightarrow e \xrightarrow{r3} e \rightarrow f \end{array}$		



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

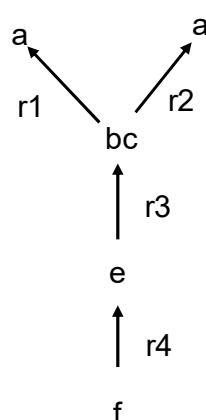
### Suy diễn lùi

Từ kết luận lùi ngược lại để tìm ra con đường đi đến lời giải cho đến khi có đủ giả thiết.

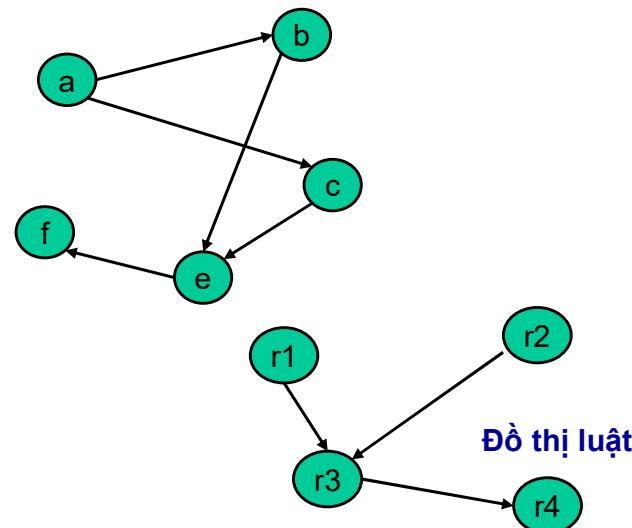
1.  $a \rightarrow b$
2.  $a \rightarrow c$
3.  $bc \rightarrow e$
4.  $e \rightarrow f$

Hỏi  $a \rightarrow f$

Kết quả:  
 $\begin{array}{l} r1 \\ a \rightarrow b \\ r2 \\ a \rightarrow c \\ r3 \\ bc \rightarrow e \xrightarrow{r4} e \rightarrow f \end{array}$



Xây dựng đồ thị sự kiện



Đồ thị luật

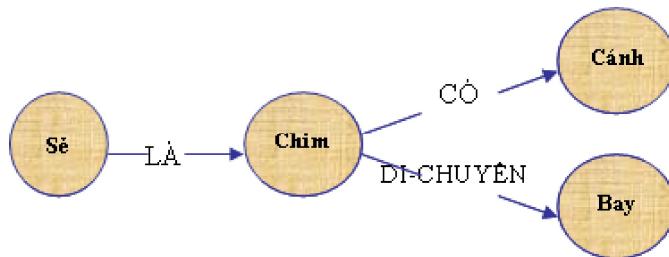
Đồ thị giúp giải quyết tương tranh, mâu thuẫn, thứ tự ưu tiên của luật



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### MẠNG NGỮ NGHĨA

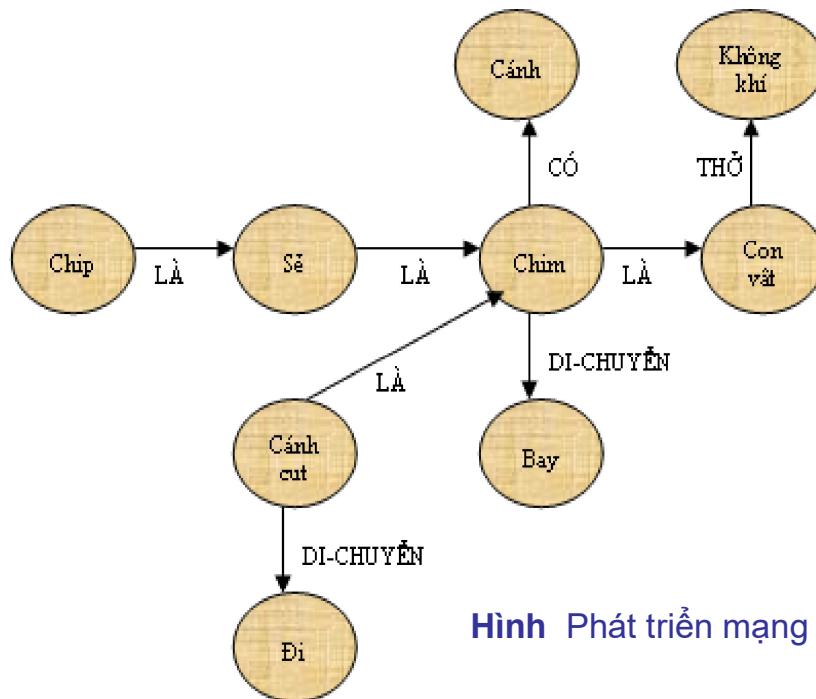
Mạng ngữ nghĩa là một phương pháp biểu diễn tri thức dùng đồ thị trong đó nút biểu diễn đối tượng và cung biểu diễn quan hệ giữa các đối tượng.



Hình "Sẻ là Chim" thể hiện trên mạng ngữ nghĩa



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC



Hình Phát triển mạng ngữ nghĩa



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Giải bài toán tam giác tổng quát

- Có 22 yếu tố của tam giác. Như vậy có  $C_{22}^3 - 1$  cách để xây dựng hay xác định một tam giác.
- Theo thống kê, có khoảng 200 công thức liên quan đến cạnh và góc 1 tam giác.
- Để giải bài toán này bằng công cụ mạng ngữ nghĩa, sử dụng khoảng 200 đỉnh để chứa công thức và khoảng 22 đỉnh để chứa các yếu tố của tam giác. Mạng ngữ nghĩa cho bài toán này có cấu trúc như sau :
  - Đỉnh của đồ thị bao gồm hai loại :
    - Đỉnh chứa công thức (ký hiệu bằng hình chữ nhật)
    - Đỉnh chứa yếu tố của tam giác (ký hiệu bằng hình tròn)
  - Cung : chỉ nối từ đỉnh hình tròn đến đỉnh hình chữ nhật cho biết yếu tố tam giác xuất hiện trong công thức nào
  - Lưu ý : trong một công thức liên hệ giữa n yếu tố của tam giác, ta giả định rằng nếu đã biết giá trị của n-1 yếu tố thì sẽ tính được giá trị của yếu tố còn lại. Chẳng hạn như trong công thức tổng 3 góc của tam giác bằng  $180^\circ$  thì khi biết được hai góc, ta sẽ tính được góc còn lại.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

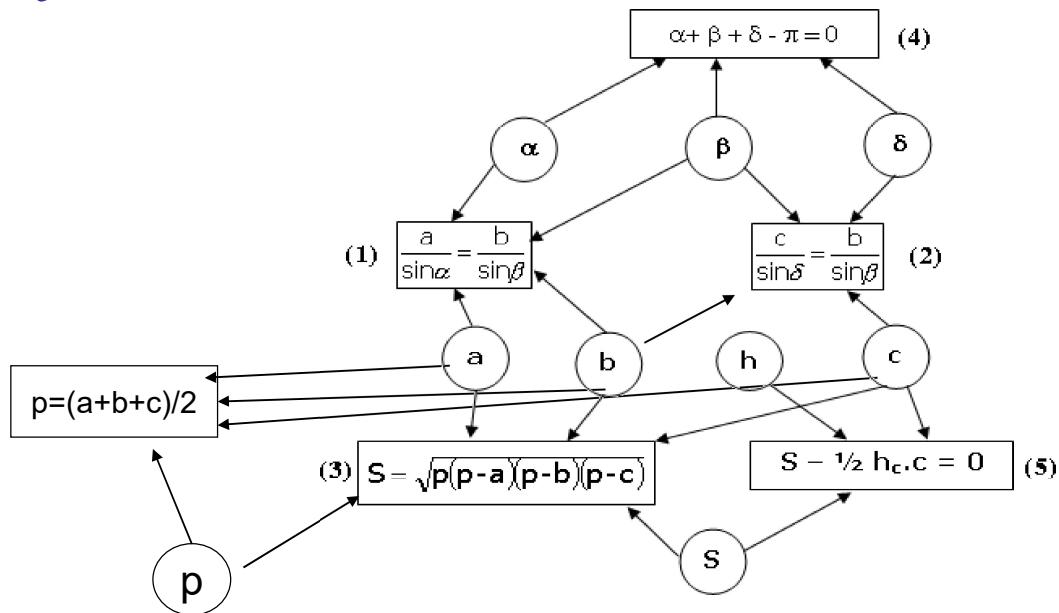
### Giải bài toán tam giác tổng quát

- B1 : Kích hoạt những đỉnh hình tròn đã cho ban đầu (những yếu tố đã có giá trị)
- B2 : Lặp lại bước sau cho đến khi kích hoạt được tất cả những đỉnh ứng với những yếu tố cần tính hoặc không thể kích hoạt được bất kỳ đỉnh nào nữa.
- Nếu một đỉnh hình chữ nhật có cung nối với n đỉnh hình tròn mà n-1 đỉnh hình tròn đã được kích hoạt thì kích hoạt đỉnh hình tròn còn lại (và tính giá trị đỉnh còn lại này thông qua công thức ở đỉnh hình chữ nhật).



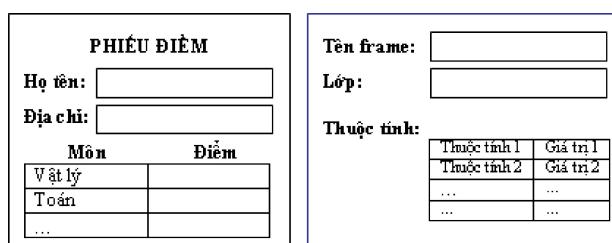
## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Ví dụ : "Cho hai góc  $\alpha, \beta$  và chiều dài cạnh  $a$  của tam giác. Tính chiều dài đường cao  $h_c$ ".

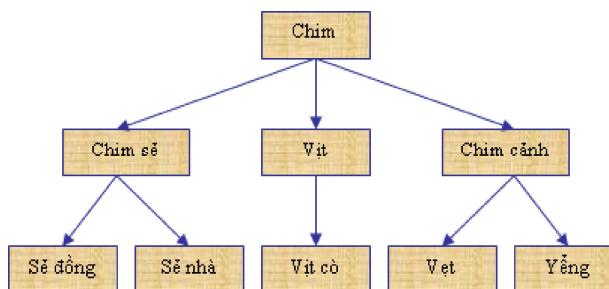


## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Frame



Hình Cấu trúc frame



Hình Nhiều mức của frame mô tả quan hệ phức tạp hơn



# BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

## LOGIC MỆNH ĐỀ VÀ LOGIC VỊ TỪ

### Logic mệnh đề

**Khái niệm:**

Mệnh đề: một phát biểu có giá trị là đúng hoặc sai.

Các luật cơ bản:

1. Luật phủ định:  $\neg(\neg P) = P$
2. Luật kéo theo:  $(P \Rightarrow Q) = (\neg P \vee Q)$
3. Luật tương phản:  $(P \Rightarrow Q) = (\neg Q \Rightarrow \neg P)$
4. Luật De Morgan:  $\neg(P \vee Q) = (\neg P \wedge \neg Q)$  và  $\neg(P \wedge Q) = (\neg P \vee \neg Q)$
5. Luật giao hoán:  $(P \wedge Q) = (Q \wedge P)$  và  $(P \vee Q) = (Q \vee P)$
6. Luật kết hợp:  $((P \wedge Q) \wedge R) = (P \wedge (Q \wedge R))$  và  $((P \vee Q) \vee R) = (P \vee (Q \vee R))$
7. Luật phân phối:  $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$  và  
 $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$



# BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Các cơ chế suy diễn

**Modus Ponens:** Nếu mệnh đề A đúng và  $A \Rightarrow B$  đúng thì  
giá trị của B sẽ là đúng.

**Modus Tollens:** Nếu mệnh đề  $A \Rightarrow B$  đúng và B sai thì  
giá trị của A sẽ là sai.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Chứng minh tính đúng đắn của phép suy diễn ( $a \rightarrow b$ ).

Thao tác biến đổi hình thức: khó đổi với con người và cài đặt được trên máy tính.

Lập bảng chân trị: độ phức tạp  $O(2^n)$

Hai phương pháp chứng minh mệnh đề có độ phức tạp chỉ  $O(n)$ :

- Thuật giải Vương Hạo.
- Thuật giải Robinson.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Thuật giải Vương Hạo

**B1:** Phát biểu lại giả thiết và kết luận theo dạng chuẩn:

$GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$

**B2:** Chuyển vẽ các  $GT_i$  và  $KL_i$  có dạng phủ định.

**B3:** Nếu  $GT_i$  có phép  $\wedge$  thì thay thế phép  $\wedge$  bằng dấu ","

Nếu  $KL_i$  có phép  $\vee$  thì thay thế phép  $\vee$  bằng dấu ","

**B4:** Nếu  $GT_i$  có phép  $\vee$  thì tách thành hai dòng con.

Nếu ở  $KL_i$  có phép  $\wedge$  thì tách thành hai dòng con.

**B5:** Một dòng được chứng minh nếu tồn tại chung một mệnh đề ở cả hai phía.

**B6:** a) Nếu một dòng không còn phép  $\wedge$  và  $\vee$  ở cả hai vế và ở 2 vế không có chung một mệnh đề thì dòng đó không được chứng minh.

b) Một vấn đề được chứng minh nếu tất cả dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Một số ví dụ:

**B2:** Ví dụ:  $p \vee q, \neg(r \wedge s), \neg g, p \vee r \rightarrow s, \neg p$   
 $\Rightarrow p \vee q, p \vee r, p \rightarrow (r \wedge s), g, s$

**B3:** Ví dụ:  $p \wedge q, r \wedge (\neg p \vee s) \rightarrow \neg q, \neg s$   
 $\Rightarrow p, q, r, \neg p \vee s \rightarrow \neg q, \neg s$

**B4:** Ví dụ:  $p, \neg p \vee q \rightarrow q$   
 $\Rightarrow p, \neg p \rightarrow q$  và  $p, q \rightarrow q$

**B5:** Ví dụ:  $p, q \rightarrow q$  được chứng minh  
 $p, \neg p \rightarrow q \Rightarrow p \rightarrow p, q$



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

- Hoạt động dựa trên:
  - + Phương pháp chứng minh phản chứng.
  - + Phép hợp giải Robinson.
- Phương pháp chứng minh phản chứng:
  - Chứng minh ( $a \rightarrow b$ ) là đúng.
  - Phản chứng: giả sử  $b$  sai, suy ra  $\neg b$  là đúng.
- Phép hợp giải Robinson:
  - i)  $p \wedge (\neg p \vee q) \rightarrow q$
  - ii)  $(p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$
- Bài toán được chứng minh nếu  $a$  đúng và  $\neg b$  đúng.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Thuật giải Robinson

**B1:** Phát biểu lại giả thiết và kết luận dưới dạng chuẩn:

$$GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$$

**B2:** Nếu  $GT_i$  có phép  $\wedge$  thì thay bằng dấu ","

Nếu  $KL_i$  có phép  $\vee$  thì thay bằng dấu ","

**B3:** Biến đổi dòng chuẩn ở B1 thành danh sách mệnh đề:

$$\{GT_1, GT_2, \dots, GT_n, \neg KL_1, \neg KL_2, \dots, \neg KL_m\}$$

**B4:** Nếu danh sách mệnh đề có 2 mệnh đề đối ngẫu nhau thì bài toán được chứng minh. Ngược lại thì chuyển sang B5.

**B6:** Áp dụng phép hợp giải: i)  $p \wedge (\neg p \vee q) \rightarrow q$

$$ii) (p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$$

**B7:** Nếu không xây dựng được thêm mệnh đề mới và danh sách mệnh đề không có 2 mệnh đề đối ngẫu thì vấn đề không được chứng minh.

Ví dụ : Chứng minh rằng

$$(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r \vee s) \wedge (\neg s \vee \neg u) \rightarrow \neg p \vee \neg u$$



# BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

## LOGIC VỊ TỪ

### Khái niệm

Vị từ và lượng từ ( $\forall$  - với mọi,  $\exists$  - tồn tại) tăng cường tính cấu trúc của mệnh đề.

Mệnh đề được biểu diễn dưới dạng :

**Vị từ ( $<\text{đối tượng } 1>, <\text{đối tượng } 2>, \dots, <\text{đối tượng } n>$ )**

(Vị từ: mỗi liên hệ giữa các đối tượng tri thức.)



# BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Định nghĩa:

- Mệnh đề là phát biểu đúng hoặc sai.
- Vị từ bậc n : phát biểu chứa n biến  $x_i$  ( $i = 1, \dots, n$ )  
 $x_i$  nhận giá trị cụ thể thì nó trở thành mệnh đề.
- Lượng từ  $\exists$  hoặc  $\forall$  đặt trước vị từ làm giảm bậc của nó 1 đơn vị.

Ví dụ:  $(x+y>2)$  : vị từ bậc 2

$(\forall x) (x+y>2)$  : vị từ bậc 1

$(\forall x) (\exists y) (x+y>2)$  : một mệnh đề.



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Giải quyết vấn đề bằng logic vị từ

Sử dụng phép thay thế và phép hợp giải Robinson.

Ví dụ: Cho vị từ  $P(x,y)$  và các mệnh đề sau:

1.  $\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)$
2.  $P(a,b)$
3.  $P(b,c)$
4.  $P(c,d)$
5.  $\neg P(a,d)$

---

Artificial Intelligence

Nguyễn Đình Cường 01



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

*Quá trình hợp giải:*

- |                               |                |                  |
|-------------------------------|----------------|------------------|
| 6. $P(a,z) \vee \neg P(b,z)$  | thể $x=a, y=b$ | hợp giải 1 và 2  |
| 7. $P(a,c)$                   | thể $z=c$      | hợp giải 3 và 6  |
| 8. $\neg P(b,d)$              | thể $z=d$      | hợp giải 5 và 6  |
| 9. $P(x,b) \vee \neg P(x,a)$  | thể $y=a, z=b$ | hợp giải 1 và 2  |
| 10. $P(b,z) \vee \neg P(c,z)$ | thể $x=b, y=c$ | hợp giải 1 và 3  |
| 11. $P(b,d)$                  | thể $z=d$      | hợp giải 4 và 10 |
| 12. Mâu thuẫn                 |                | hợp giải 8 và 11 |

---

Artificial Intelligence

Nguyễn Đình Cường 02



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

---

### **Bài toán 1: Giải bài toán tìm đường đi**

Định nghĩa 2 vị từ:

1.  $P(x,y)$  : có đường đi trực tiếp (road) từ  $x$  đến  $y$
2.  $Q(x,y)$  : có đường đi (route) từ  $x$  đến  $y$

Các mệnh đề:

1.  $(\exists x)(\exists y) (P(x,y))$
2.  $(\forall x) (\forall y) (P(x,y) \rightarrow Q(x,y))$
3.  $(\forall x) (\forall y) (\exists z) (Q(x,z) \wedge P(z,y) \rightarrow Q(x,y))$



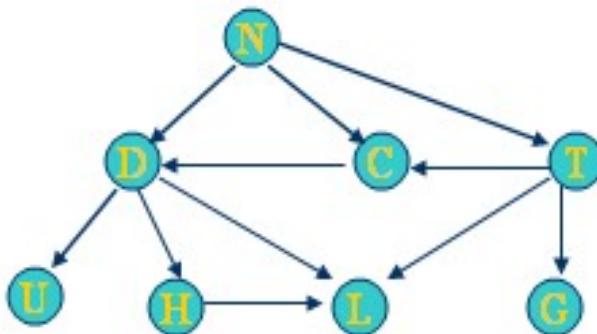
## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

---

### **Giải bài toán tìm đường đi**

Ví dụ: Cho đồ thị, tìm đường đi từ N đến L.

1.  $P(N,C)$
2.  $P(C,D)$
3.  $P(N,T)$
4.  $P(N,D)$
5.  $P(T,G)$
6.  $P(T,L)$
7.  $P(T,C)$
8.  $P(D,U)$
9.  $P(D,H)$
10.  $P(H,L)$
11.  $P(D,L)$
12.  $\neg P(x,y) \vee Q(x,y))$  : Có đường đi trực tiếp từ  $x \rightarrow y$  thi cung xem như có đường đi từ  $x \rightarrow y$
13.  $\neg Q(x,f(x,y)) \vee \neg P(f(x,y),y) \vee Q(x,y))$  : Nếu có đường đi từ  $x \rightarrow$  đến một đỉnh trung gian đi từ  $x$ :  $f(x,y)$  và từ đỉnh trung gian này có đường đi trực tiếp đến  $y$ , thì ta có đường đi từ  $x \rightarrow y$
14.  $\neg Q(N,L)$  Hỏi có đường đi từ :  $N \rightarrow L$  ?





## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Giải bài toán tìm đường đi

Ví dụ: Cho đồ thị, tìm đường đi từ N đến L. (tt)

Quá trình hợp giải:

- |  |                      |
|--|----------------------|
| 15. Q(N,C)      thế $x=N, y=C$           | hợp giải 1 và 12     |
| 16. Q(N,D)      thế $x=N, f(x,y)=C, y=D$ | hợp giải 2,13 và 15  |
| 17. Q(N,L)      thế $x=N, f(x,y)=D, y=L$ | hợp giải 11,13 và 16 |
| 18. Có điều phải chứng minh              | hợp giải 14 và 17    |



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

### Bài toán 2: Bài toán con khỉ và nải chuối

Trong phòng có con khỉ, chiếc ghế và nải chuối...

Có thể giải bài toán nhờ vào các **vị từ**:

- |                    |                                  |
|--------------------|----------------------------------|
| 1. in_room(X)      | : có X trong phòng               |
| 2. dexterous(X)    | : X là loài khéo léo             |
| 3. tall(X)         | : X thì cao                      |
| 4. can_climb(X,Y)  | : X có thể leo lên Y             |
| 5. can_move(X,Y,Z) | : X có thể di chuyển Y đến gần Z |
| 6. get_on(X,Y)     | : X đứng trên Y                  |
| 7. under(X,Y)      | : X ở dưới Y                     |
| 8. close(X,Y)      | : X ở gần Y                      |
| 9. can_reach(X,Y)  | : X có thể cầm lấy Y             |



## BIỂU DIỄN VÀ SUY LUẬN TRÍ THỨC

Các **mệnh đề** giúp giải bài toán:

1. in\_room(bananas)
2. in\_room(chair)
3. in\_room(monkey)
4. dexterous(monkey)
5. tall(chair)
6. can\_move(mokey,chair,bananas)
7. can\_climb(monkey,chair)
8. can\_climb(X,Y) → get\_on(X,Y)
9. in\_room(X) ∧ in\_room(Y) ∧ in\_room(Z) ∧ can\_move(X,Y,Z) → under(Y,Z)
10. get\_on(X,Y) ∧ under(Y,Z) ∧ tall(Y) → close(X,Z)
11. dexterous(X) ∧ close(X,Y) → can\_reach(X,Y)

---

Artificial Intelligence

Nguyễn Đình Cường 07

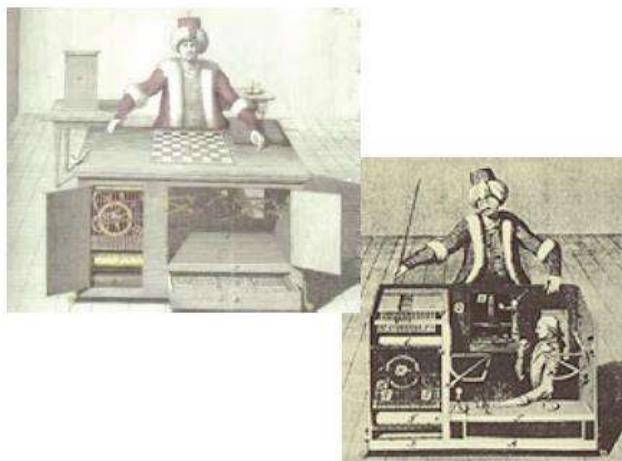


TRƯỜNG ĐẠI HỌC NHA TRANG

---

Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

Bộ môn Công Nghệ Phần Mềm

Khoa Công Nghệ Thông Tin

Nha trang - 2012

---

Artificial Intelligence

Nguyễn Đình Cường 08



# MÁY HỌC

## MỞ ĐẦU

Các chương trước đã thảo luận về biểu diễn và suy luận tri thức. Trong trường hợp này giả định đã có sẵn tri thức và có thể biểu diễn tường minh tri thức.

Tuy vậy trong nhiều tình huống, sẽ không có sẵn tri thức như:

- Kỹ sư tri thức cần thu nhận tri thức từ chuyên gia lĩnh vực.
- Cần biết các luật mô tả lĩnh vực cụ thể.
- Bài toán không được biểu diễn tường minh theo luật, sự kiện hay các quan hệ.

Có hai tiếp cận cho hệ thống học:

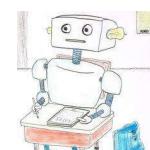
- Học từ ký hiệu: bao gồm việc hình thức hóa, sửa chữa các luật tường minh, sự kiện và các quan hệ.
- Học từ dữ liệu số: được áp dụng cho những hệ thống được mô hình dưới dạng số liên quan đến các kỹ thuật nhằm tối ưu các tham số. Học theo dạng số bao gồm mạng Neural nhân tạo, thuật giải di truyền, bài toán tối ưu truyền thống. Các kỹ thuật học theo số không tạo ra CSTT tường minh.



# MÁY HỌC

## CÁC HÌNH THỨC HỌC

1. **Học vẹt**: Hệ tiếp nhận các khẳng định của các quyết định đúng. Khi hệ tạo ra một quyết định không đúng, hệ sẽ đưa ra các luật hay quan hệ đúng mà hệ đã sử dụng. Hình thức học vẹt nhằm cho phép chuyên gia cung cấp tri thức theo kiểu tương tác.
2. **Học bằng cách chỉ dẫn**: Thay vì đưa ra một luật cụ thể cần áp dụng vào tình huống cho trước, hệ thống sẽ được cung cấp bằng các chỉ dẫn tổng quát. Ví dụ: "gas hầu như bị thoát ra từ van thay vì thoát ra từ ống dẫn". Hệ thống phải tự mình đề ra cách biến đổi từ trừu tượng đến các luật khả dụng.
3. **Học bằng qui nạp**: Hệ thống được cung cấp một tập các ví dụ và kết luận được rút ra từ từng ví dụ. Hệ liên tục lọc các luật và quan hệ nhằm xử lý từng ví dụ mới.

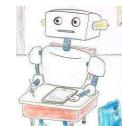




# MÁY HỌC

## CÁC HÌNH THỨC HỌC

4. **Học bằng tương tự:** Hệ thống được cung cấp đáp ứng đúng cho các tác vụ tương tự nhưng không giống nhau. Hệ thống cần làm thích ứng đáp ứng trước đó nhằm tạo ra một luật mới có khả năng áp dụng cho tình huống mới.
5. **Học dựa trên giải thích:** Hệ thống phân tích tập các lời giải ví dụ (và kết quả) nhằm ấn định khả năng đúng hoặc sai và tạo ra các giải thích dùng để hướng dẫn cách giải bài toán trong tương lai.
6. **Học dựa trên tình huống:** Bất kỳ tính huống nào được hệ thống lập luận đều được lưu trữ cùng với kết quả cho dù đúng hay sai. Khi gặp tình huống mới, hệ thống sẽ làm thích nghi hành vi đã lưu trữ với tình huống mới.
7. **Khám phá hay học không giám sát:** Thay vì có mục tiêu tường minh, hệ khám phá liên tục tìm kiếm các mẫu và quan hệ trong dữ liệu nhập. Các ví dụ về học không giám sát bao gồm gom cụm dữ liệu, học để nhận dạng các đặc tính cơ bản như cạnh từ các điểm ảnh.



Artificial Intelligence

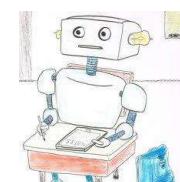
Nguyễn Đình Cường111



# MÁY HỌC

Ví dụ

- Hệ MYCIN
- Mạng Neural nhân tạo
- Thuật toán học Quinland
- Bài toán nhận dạng
- Máy chơi cờ carô, cờ tướng



Artificial Intelligence

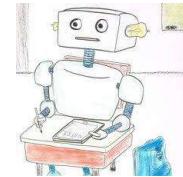
Nguyễn Đình Cường112



# MÁY HỌC

## THUẬT GIẢI Quinlan

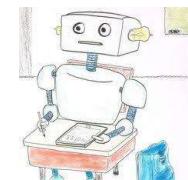
- Là thuật toán học theo quy nạp dùng luật, đa mục tiêu.
- Do Quinlan đưa ra năm 1979.
- Ý tưởng: Chọn thuộc tính quan trọng nhất để tạo cây quyết định.
- Thuộc tính quan trọng nhất là thuộc tính phân loại Bảng quan sát thành các bảng con sao cho từ mỗi bảng con này dễ phân tích để tìm quy luật chung.



# MÁY HỌC

## THUẬT GIẢI A. Quinlan

STT	Size	Nationality	Family	Conclusion
1	Small	German	Single	A
2	Large	French	Single	A
3	Large	German	Single	A
4	Small	Italian	Single	B
5	Large	German	Married	B
6	Large	Italian	Single	B
7	Large	Italian	Married	B
8	Small	German	Married	B





# MÁY HỌC

Xét vector V: có số chiều bằng số phân loại

- $V(\text{Size}=\text{Small}) = (\text{ASmall}, \text{BSmall})$
- $\text{ASmall} = \text{Số quan sát A có Size là Small / Tổng số quan sát có Size=Small}$
- $\text{BSmall} = \text{Số quan sát B có Size là Small / Tổng số quan sát có Size=Small}$ 
  - $V(\text{Size}=\text{Small}) = (1/3, 2/3)$
  - $V(\text{Size}=\text{Large}) = (2/5, 3/5)$

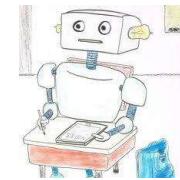
Với thuộc tính Nationality

- $V(\text{Nat} = \text{German}) = (2/4, 2/4)$
- $V(\text{Nat} = \text{French}) = (1, 0)$
- $V(\text{Nat} = \text{Italian}) = (0, 1)$

**Chọn thuộc tính có nhiều vecto đơn vị**

Thuộc tính Family:

- $V(\text{Family}=\text{Single}) = (3/5, 2/5)$
- $V(\text{Family} = \text{Married}) = (0, 1)$



# MÁY HỌC

## VỚI MỌI THUỘC TÍNH CỦA BẢNG QUAN SÁT

### Chỉ còn xét German

• Thuộc tính Size:

$$V_{(\text{Size}=\text{Small})} = (1/2, 1/2)$$

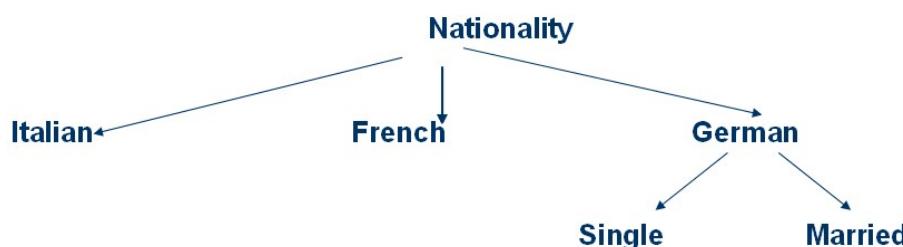
$$V_{(\text{Size}=\text{Large})} = (1/2, 1/2)$$

• Thuộc tính Family:

$$V_{(\text{Family}=\text{Single})} = (1, 0)$$

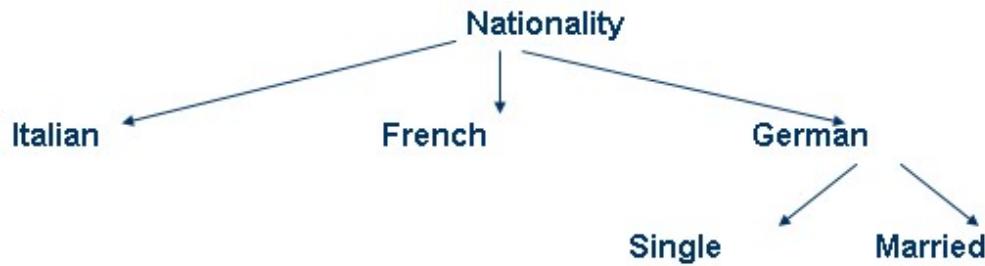
$$V_{(\text{Family}=\text{Married})} = (0, 1)$$

STT	Size	Family	Conclusion
1	Small	Single	A
2	Large	Single	A
3	Large	Married	B
4	Small	Married	B





## MÁY HỌC



**Rule 1: If (Nationality IS Italian) then (Conclusion IS B)**

**Rule 2: If (Nationality IS French) then (Conclusion IS A)**

**Rule 3: If (Nationality IS German) AND (Family IS Single)  
then (Conclusion IS A)**

**Rule 4: If (Nationality IS German) AND (Family IS Married)  
then (Conclusion IS B)**



## MÁY HỌC

### Thuật giải Học theo độ bất định

Stt	Age	Competition	Type	Profit
1	Old	No	Software	Down
2	Midle	Yes	Software	Down
3	Midle	No	Hardware	Up
4	Old	No	Hardware	Down
5	New	No	Hardware	Up
6	New	No	Software	Up
7	Midle	No	Software	Up
8	New	Yes	Software	Up
9	Midle	Yes	Hardware	Down
10	Old	Yes	Hardware	Down



## MÁY HỌC

Độ bất định của X:

$$E(X) = -\sum_{i=1}^k p_i \log_2 p_i$$

Tính Entropy cho mỗi thuộc tính và chọn thuộc tính có Entropy nhỏ nhất.

$$E(C / A) = -\sum_{i=1}^k p(c_i, a_i) \log_2 p(c_i, a_i)$$

$$E(C / \text{Competitio n= No}) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.918$$

$$E(C / \text{Competitio n= Yes}) = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} = 0.811$$

$$E(C / \text{Competitio n}) = 0.6 * 0.918 + 0.4 * 0.811 = 0.8752$$



## MÁY HỌC

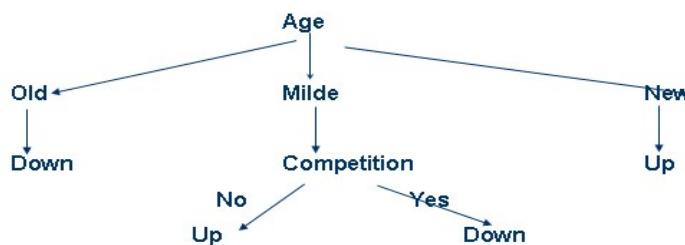
Tương tự:

$$E(C/Age) = 0.4$$

$$E(C/Type) = 0.97024$$

Age cho nhiều thông tin nhất

STT	Competition	Type	Profit
1	Yes	Software	Down
2	No	Hardware	Up
3	No	Software	Up
4	Yes	Hardware	Down



Rule 1: If (Age IS Old) then (Profit IS Down)

Rule 2: If (Age IS New) then (Profit IS Up)

Rule 3: If (Age IS Middle) And (Competition IS No)  
then (Profit IS Up)

Rule 4: If (Age IS Middle) And (Competition IS Yes)  
then (Profit IS Down)



# MÁY HỌC



Artificial Intelligence

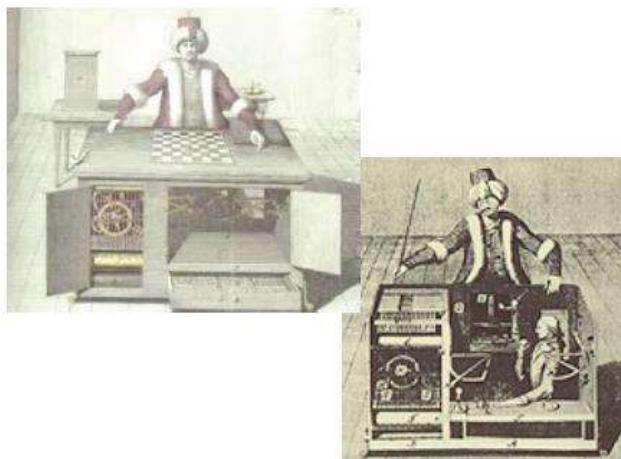
Nguyễn Đình Cường 21



TRƯỜNG ĐẠI HỌC NHA TRANG

Bài giảng

## TRÍ TUỆ NHÂN TẠO



NGUYỄN ĐÌNH CƯỜNG

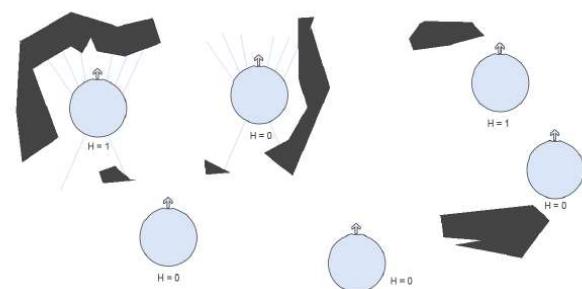
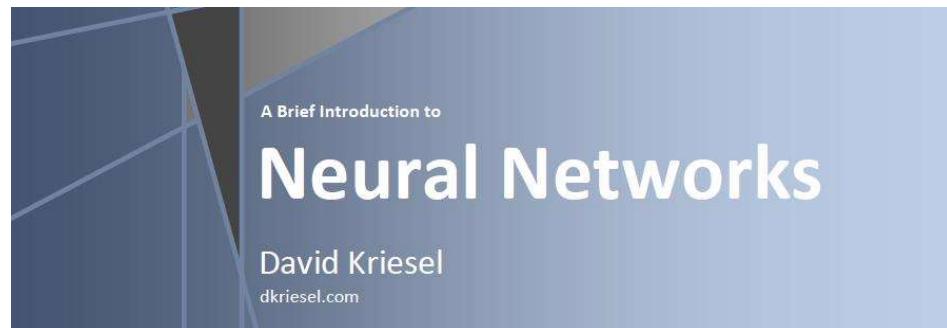
Bộ môn Công Nghệ Phần Mềm

Khoa Công Nghệ Thông Tin

Nha trang - 2012

Artificial Intelligence

Nguyễn Đình Cường 22



From biology to formalization –  
motivation, philosophy, history and  
realization of neural models

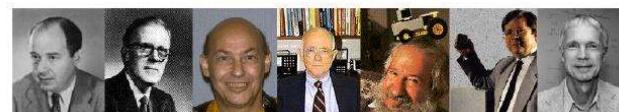
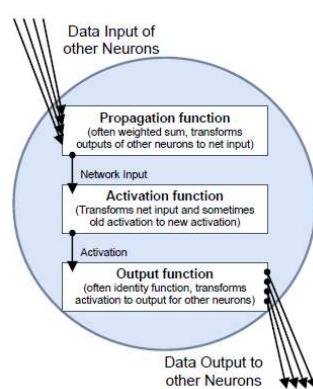
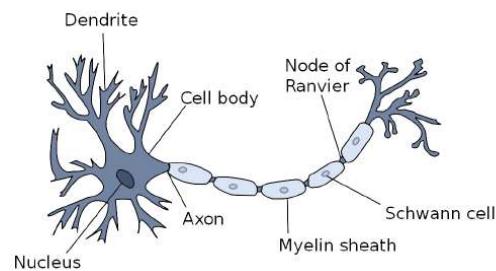
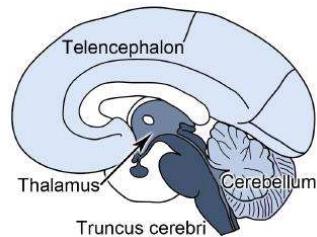
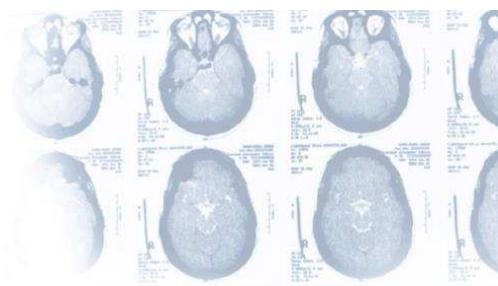


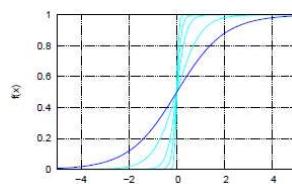
Figure 1.4: Some institutions of the field of neural networks. From left to right: John von Neumann, Donald O. Hebb, Marvin Minsky, Bernard Widrow, Seymour Papert, Teuvo Kohonen, John Hopfield, "in the order of appearance" as far as possible.

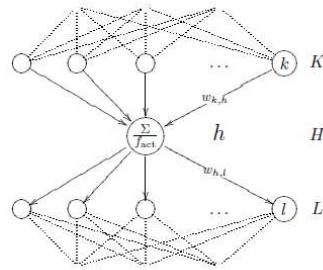
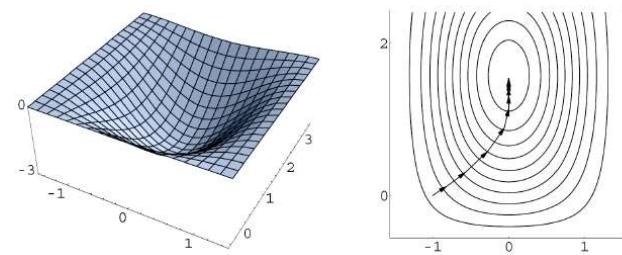


$$net_j = f_{\text{prop}}(o_{i1}, \dots, o_{in}, w_{i1,j}, \dots, w_{in,j})$$

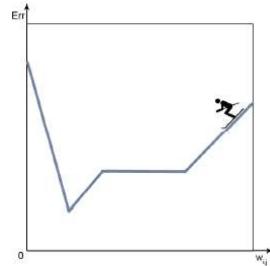
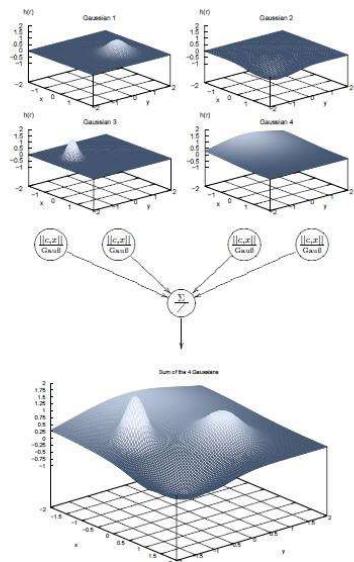
$$\text{net}_j = \sum_{i \in I} (o_i \cdot w_{i,j}) \cdot \frac{1}{1 + e^{-x}},$$

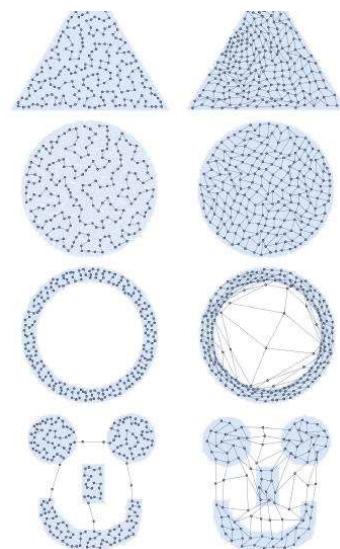
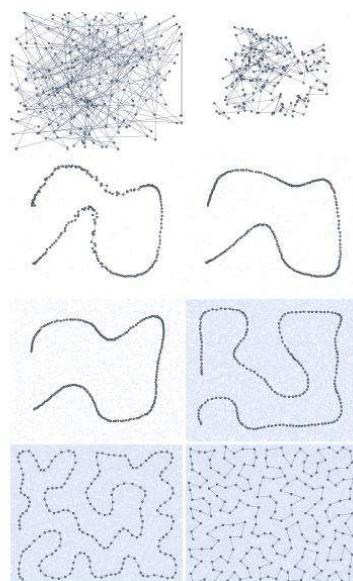
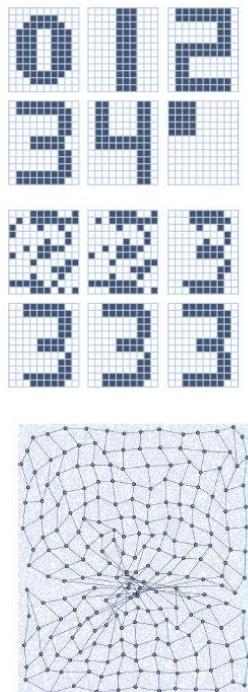
$$\frac{1}{1 + e^{-\frac{x}{T}}}.$$





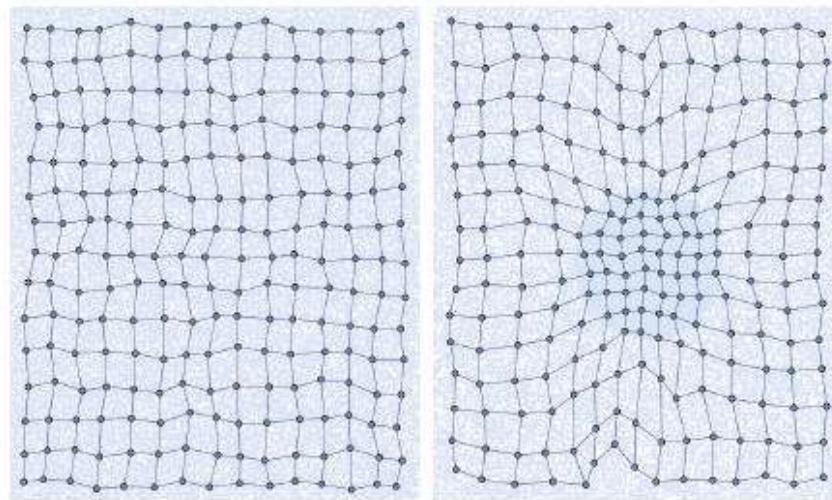
$$\Delta w_{i,j}(t) = \eta o_i \delta_j + \alpha \cdot \Delta w_{i,j}(t - 1)$$





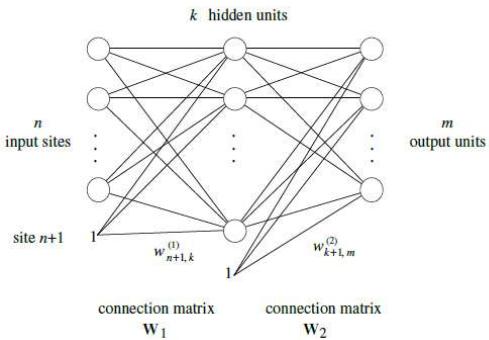
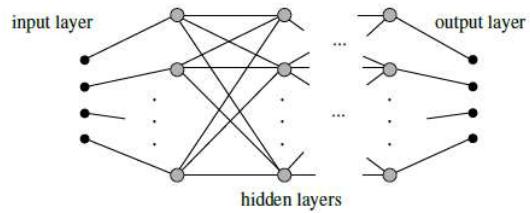
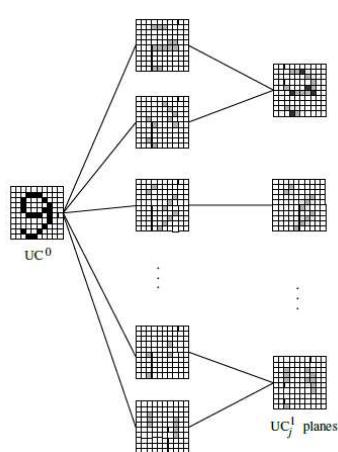
Artificial Intelligence

Nguyễn Đình Cường 29



Artificial Intelligence

Nguyễn Đình Cường 30



OpenCV 2.4.13.6 documentation » OpenCV API Reference » ml Machine Learning » [previous](#) | [next](#) | [index](#)

## Neural Networks

ML implements feed-forward artificial neural networks or, more particularly, multi-layer perceptrons (MLP), the most commonly used type of neural networks. MLP consists of the input layer, output layer, and one or more hidden layers. Each layer of MLP includes one or more neurons directionally linked with the neurons from the previous and the next layer. The example below represents a 3-layer perceptron with three inputs, two outputs, and the hidden layer including five neurons:

All the neurons in MLP are similar. Each of them has several input links (it takes the output values from several neurons in the previous layer as input) and several output links (it passes the response to several neurons in the next layer). The values retrieved from the previous layer are summed up with certain weights, individual for each neuron, plus the bias term. The sum is transformed using the activation function  $\sigma$  that may be also different for different neurons.



```
915 // initialize training data
916 Mat inputs = trainData->getTrainSamples();
917 Mat outputs = trainData->getTrainResponses();
918 Mat sw = trainData->getTrainSampleWeights();
919 prepare_to_train( inputs, outputs, sw, flags );
920
921 // ... and link weights
922 if( !(flags & UPDATE_WEIGHTS) )
923     init_weights();
924
925 TermCriteria termcrit;
926 termcrit.type = TermCriteria::COUNT + TermCriteria::EPS;
927 termcrit.maxCount = std::max((params.termCrit.type & CV_TERMCRIT_ITER ? params.termCrit.maxCount : MAX_ITER), 1);
928 termcrit.epsilon = std::max((params.termCrit.type & CV_TERMCRIT_EPS ? params.termCrit.epsilon : DEFAULT_EPSILON), DBL_EPSILON);
929
930 int iter = 0;
931 switch(params.trainMethod){
932 case ANN_MLP::BACKPROP:
933     iter = train_backprop(inputs, outputs, sw, termcrit);
934     break;
935 case ANN_MLP::RPROP:
936     iter = train_rprop(inputs, outputs, sw, termcrit);
937     break;
938 case ANN_MLP::ANNEAL:
939     iter = train_anneal(trainData);
940     break;
941 }
942 trained = iter > 0;
943 return trained;
944 }
945 int train_anneal(const Ptr<TrainData>& trainData)
946 {
947     SimulatedAnnealingANN_MLP s(*this, trainData);
```



```
1 #include <opencv2/ml/ml.hpp>
2
3 // #include <ml.hpp>
4
5 using namespace std;
6 using namespace cv;
7 using namespace cv::ml;
8
9 int main()
10 {
11     // create random training data
12     Mat<float> data(100, 100);
13     randn(data, Mat::zeros(1, 1, data.type()), Mat::ones(1, 1, data.type()));
14
15     // half of the samples for each class
16     Mat<float> responses(data.rows, 2);
17     for (int i = 0; i<data.rows; ++i)
18     {
19         if (i < data.rows/2)
20         {
21             responses(i, 0) = 1;
22             responses(i, 1) = 0;
23         }
24         else
25         {
26             responses(i, 0) = 0;
27             responses(i, 1) = 1;
28         }
29     }
30
31     /* example code for just a single response (regression)
32     Mat<float> responses(data.rows, 1);
```

C:\opencv\sources\samples\cpp>cl /I C:\opencv\build\include neural\_network.cpp /link  
opencv\_world341d.lib opencv\_world341.lib



## TRƯỜNG ĐẠI HỌC NHA TRANG

A screenshot of a Windows desktop environment showing a Visual Studio Code window. The terminal tab is active, displaying the output of a C++ program named 'neural\_network.cpp'. The program uses OpenCV's neural network samples to predict values from a dataset. The output shows numerous predictions as pairs of floats, such as [0.49815065, 0.49921253], [0.49815512, 0.4992235], etc. The code in the editor shows the main function and a loop for processing rows of data.

```
neural_network.cpp - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
neural_network.cpp •
x64 Native Tools Command Prompt for VS 2017
C:\opencv\sources\samples\cpp>neural_network.exe
Predict one-vector:
[0.49815065, 0.49921253]
Predict training data:
[0.49815512, 0.4992235]
[0.49815148, 0.4992148]
[0.49816421, 0.49922886]
[0.49815488, 0.49921444]
[0.4981553, 0.49921337]
[0.49815768, 0.4992235]
[0.49815473, 0.49921861]
[0.49815747, 0.49921554]
[0.49815911, 0.49922082]
[0.4981544, 0.49921104]
[0.49815312, 0.49920845]
[0.49815834, 0.4992134]
[0.4981578, 0.499221]
[0.49815252, 0.49921796]
[0.49815798, 0.49920991]
[0.4981553, 0.49921247]
[0.49815372, 0.49922264]
[0.49815732, 0.49922493]
[0.49815607, 0.49922168]
[0.49814662, 0.49921648]
[0.49815735, 0.49921829]
[0.49815637, 0.4992227]
[0.49815381, 0.49921602]
[0.49815139, 0.49921939]
[0.49815428, 0.49921471]
[0.4981533, 0.49922156]
/*
 *example code for just a single response (regression)
 Mat<float> responses(data.rows, 1);
 for (int i=0; i<responses.rows; ++i)
 | responses(i, 0) = i < responses.rows / 2 ? 0 : 1;
 */
main() Ln 63, Col 6 Spaces:4 UTF-8 CRLF C++ Win32 3:36 PM 5/22/2018
```

Artificial Intelligence

Nguyễn Đình Cường 35



## TRƯỜNG ĐẠI HỌC NHA TRANG

[1] Neural network, Raul Rojas, 1996.

[2] Neural network, David Kriesel, 2005

[3] Bài Giảng Trí Tuệ Nhân Tạo, Nguyễn Đình Thuân-Đại học Nha Trang.

[3] Opencv Lib

Artificial Intelligence

Nguyễn Đình Cường 36