CHƯƠNG 3 - KỸ THUẬT GIẢI QUYẾT VẤN ĐỀ

NỘI DUNG

Vấn đề - giải quyết vấn đề

Các phương pháp biểu diễn vấn đề

Tìm kiếm lời giải trên không gian bài toán

Tìm kiếm có đối thủ

Các phương pháp khác

3.1. GIẢI QUYẾT VẤN ĐỀ VÀ TRÍ TUỆ NHÂN TẠO

- Giải quyết vấn đề của con người là một trường hợp riêng của quá trình xử lý thông tin trong bộ não: tìm cách đi từ tình huống ban đầu nào đó để đến một đích mong muốn (*tìm kiếm*, *tính*)
- Các chiến lược:

  Ước lượng mức độ phức tạp của vấn đề: (i) nếu đơn giản, (ii) nếu phức tạp ... Biểu diễn - xử lý

  Có thể nới lỏng ràng buộc của bài toán

  Có thể giải theo phương pháp thử và sai

HÌNH THỨC HÓA BÀI TOÁN

- Quá trình chuyển từ bài toán sang cách biểu diễn nhờ ký hiệu: dạng biểu diễn (~ tình huống, trạng thái), mẫu liên kết (~ phép biến đổi)
  → Không gian bài toán
- Giải quyết vấn đề là quá trình xuất phát từ dạng ban đầu và tìm kiếm trong không gian bài toán để tìm ra dãy các phép toán (biến đổi, hành động) cho phép dẫn tới đích mong muốn.

PHÂN LOẠI VẤN ĐỀ

- Không gian bài toán: trạng thái đầu, đích, phép biến đổi
- Bài toán phát biểu chỉnh: biết rõ đầu vào, đầu ra và quyết định được, khi nào bài toán được coi là giải quyết. Với mỗi lời giải giả định có thể áp dụng thuật toán để xác định có phải là lời giải ...
- Bài toán phát biểu không chỉnh: ngược lại (đích không tường minh, các toán tử không được chỉ ra, không gian bài toán không hữu hạn, ràng buộc thời gian ...)

PHÂN LOẠI VẤN ĐỀ

BT phát biểu chỉnh

BT phát biểu không chỉnh

ĐPT đa thức

$O(n^{\alpha})$

ĐPT hàm mũ

$O(\alpha^{n})$

giải được

không giải được

Giải thuật

Mẹo giải

VÍ DỤ 1: BÀI TOÁN ĐỐ CHỮ

- Hãy thay các chữ cái bằng các chữ số từ 0 đến 9 sao cho không có hai chữ cái nào được thay bởi cùng một số và thỏa mãn ràng buộc sau:

SEND	CROSS
+ MORE	+ ROADS
MONEY	DANGER

$$\begin{array}{r} \text{SEND} \\ +\ \text{MORE} \\ \hline \text{MONEY} \end{array}$$

AL = {D,E,M,N,O,R,S,Y}, DI = {0,1,...,9}

Tìm f: AL$\rightarrow$DI, sao cho $\forall x \neq y$ thì $f(x) \neq f(y)$

(*máy tính: vét cạn, người: mẹo*)

$D+E = Y + 10.C1$ (1)

$N+R+C1 = E + 10.C2$ (2)

$E+O+C2 = N + 10.C3$ (3)

$S+M+C3 = O + 10.C4$ (4)

$M = C4$ (5)

$C1,C2,C3,C4 \in \{0,1\}$ (6)

$S \neq 0, M \neq 0$ (7)

Các chữ cái khác nhau (8)

$$\begin{array}{r} 9567 \\ +\ 1085 \\ \hline 10652 \end{array}$$

(5,6,7) $\Rightarrow$ M = 1, C4 = 1

(4) $\Rightarrow$ S+1+C3 = O+10 $\Rightarrow$ O+10 $\leq$ 9+1+1

O = 1 = M (loại)

O = 0, S+C3 = 9

S+C3 = 9 $\Rightarrow$

C3=1, S=8 $\Rightarrow$ (3) E+C2 = 10+N $\Rightarrow$ N=0 loại

C3=0, S=9 $\Rightarrow$ E+C2 = N $\Rightarrow$ C2=1, E+1 = N

(2) $\Rightarrow$ N+R+C1 = E+10 $\Rightarrow$ R+C1=9 $\Rightarrow$ R=8

(1) $\Rightarrow$ D+E=Y+10, E$\leq$6 $\Rightarrow$ 2$\leq$Y$\leq$3

Y=3 $\Rightarrow$ D=7, E=6, N=7 loại

Y=2 $\Rightarrow$ D=7, E=5, N=6

VÍ DỤ 2: BÀI TOÁN RÓT NƯỚC

- Cho hai bình A(m lít), B(n lít), làm cách nào để đong được k lít ($k \le max(m,n)$) chỉ bằng hai bình A, B và một bình trung gian C.

  Các thao tác rót (how):

  C → A; C → B; A → B; A → C; B → A; B → C

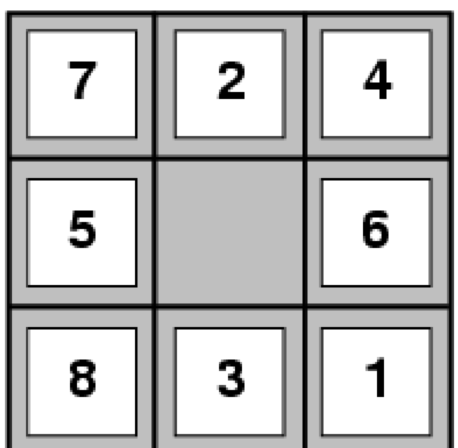
  Điều kiện: không tràn, đổ hết
- Ví dụ: m = 5, n = 6, k = 2 (what)
- Mô hình toán học:

  (x, y) → (x', y')

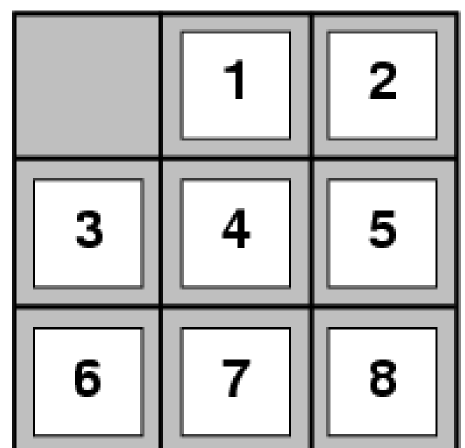
  A B A B

VÍ DỤ 3: TRÒ CHƠI n^2-1 SỐ

- Trong bảng ô vuông n hàng, n cột, mỗi ô chứa một số nằm trong phạm vi từ 1 → n^2-1 sao cho không có hai ô có cùng giá trị, còn đúng một ô bị trống. Xuất phát từ một cách sắp xếp nào đó của các số trong bảng, hãy dịch chuyển các ô trống sang phải, sang trái, lên trên, xuống dưới để đưa về bảng:



Start State



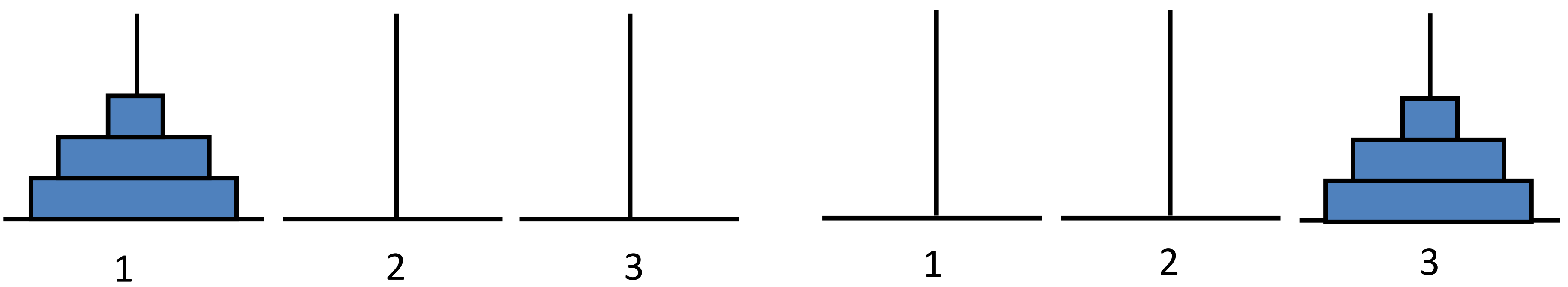
Goal State

VÍ DỤ 4: BÀI TOÁN THÁP HÀ NỘI

Cho ba cọc 1,2,3. Ở cọc 1 ban đầu có n đĩa, sắp theo thứ tự to dần từ trên xuống dưới. Hãy tìm cách chuyển n đĩa đó sang cọc 3 sao cho:

. Mỗi lần chỉ chuyển một đĩa

. Ở mỗi cọc không cho phép đĩa to (C) nằm trên đĩa con (A)



Bài toán tháp Hà Nội với n = 3

VÍ DỤ 5: BÀI TOÁN QUAN TÒA - HỀ - KẺ TRỘM

- Có ba người ngồi quanh một bàn tròn. Một người qua đường nghe thấy ba người này nói chuyện với nhau:
  - người 1 nói 2 là quan tòa
  - người 2 nói 3 là hề
  - người 3 nói 1 là trộm
- Biết rằng: hề luôn nói đùa; quan tòa nói thật; trộm nói dối
- Hỏi ai là ai?

VÍ DỤ 6 - TÍNH TUỔI 3 CON CỦA ÔNG M

Ông M gặp người bạn là nhà toán học, ông nói:

- Hôm nay là một ngày đặc biệt, sinh nhật của cả 3 con tôi, anh có biết chúng bao nhiêu tuổi không ?
- Chắc rồi, anh hãy nói về các con anh đi
- Tôi tiết lộ nhé: tích tuổi 3 con tôi là 36
- Tuyệt vời, tôi cần biết thêm nữa !
- Tổng tuổi của chúng bằng số cửa sổ của tòa nhà kia kìa.

Nhà toán học nghĩ và vẫn cần thêm thông tin.

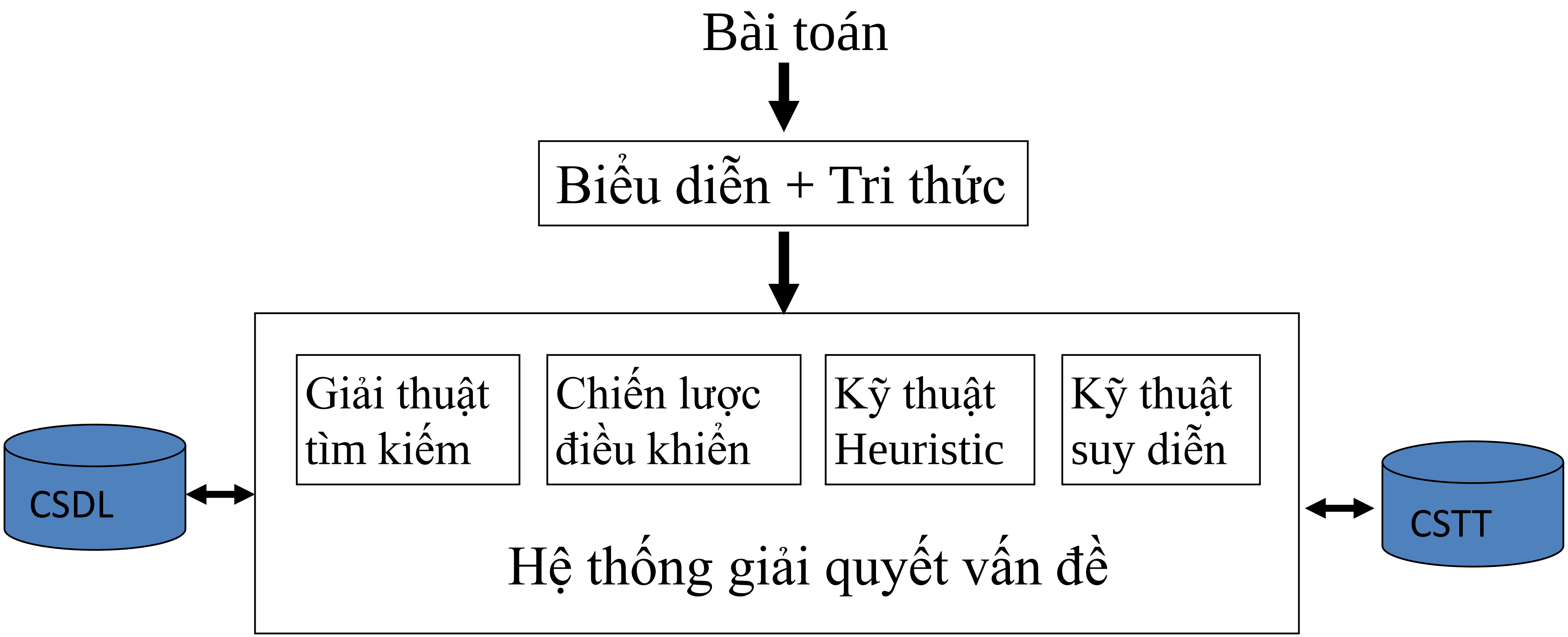
- Ông M: Con trai lớn nhất của tôi có mắt xanh

Nhà toán học nói đúng tuổi !

CÁC ĐẶC TRƯNG CƠ BẢN CỦA VẤN ĐỀ

- Bài toán có thể phân rã?
- Không gian bài toán có thể đoán trước?
- Có tiêu chuẩn xác định lời giải tối ưu?
- Có cơ sở tri thức phi mâu thuẫn?
- Tri thức cần cho quá trình tìm kiếm hay để điều khiển?
- Có cần tương tác người – máy?

NHỮNG YẾU TỐ CƠ BẢN TRONG GIẢI QUYẾT VẤN ĐỀ



3.2. CÁC PHƯƠNG PHÁP BIỂU DIỄN BÀI TOÁN

- Biểu diễn nhờ Không gian trạng thái
- Qui về các bài toán con
- Sử dụng logic hình thức
- Biểu diễn bài toán cho máy tính

BIỂU DIỄN BÀI TOÁN

Biểu diễn nhờ KGTT

- Mỗi hình trạng của bài toán tương ứng với một trạng thái (state)
- Mỗi phép biến đổi từ hình trạng này sang hình trạng khác tương ứng với các toán tử (operator)

Qui bài toán về bài toán con

- Phân chia bài toán thành các bài toán con, các bài toán con lại được phân rã tiếp cho đến khi gặp được các bài toán sơ cấp cho phép xác định lời giải của bài toán ban đầu trên cơ sở lời giải của các bài toán con.

BIỂU DIỄN BÀI TOÁN

Sử dụng logic hình thức

Khi giải quyết bài toán, có thể tiến hành dẫn xuất logic để thu gọn quá trình tìm kiếm, nhiều khi chứng minh được không có lời giải.

(logic mệnh đề, logic vị từ bậc nhất)

cho phép:

- Kiểm tra điều kiện kết thúc trong tìm kiếm đối với không gian trạng thái, kiểm tra tính áp dụng được của các toán tử
- Chứng minh không tồn tại lời giải
- Mục đích: chứng minh một phát biểu nào đó trên cơ sở những tiền đề và luật suy diễn đã có.

BIỂU DIỄN TRONG MÁY TÍNH

Dùng bảng/mảng (array): ví dụ, trò chơi n^2-1 số

Trạng thái đầu

11	14	4	7
10	6		5
1	2	13	15
9	12	8	3

Trạng thái đích

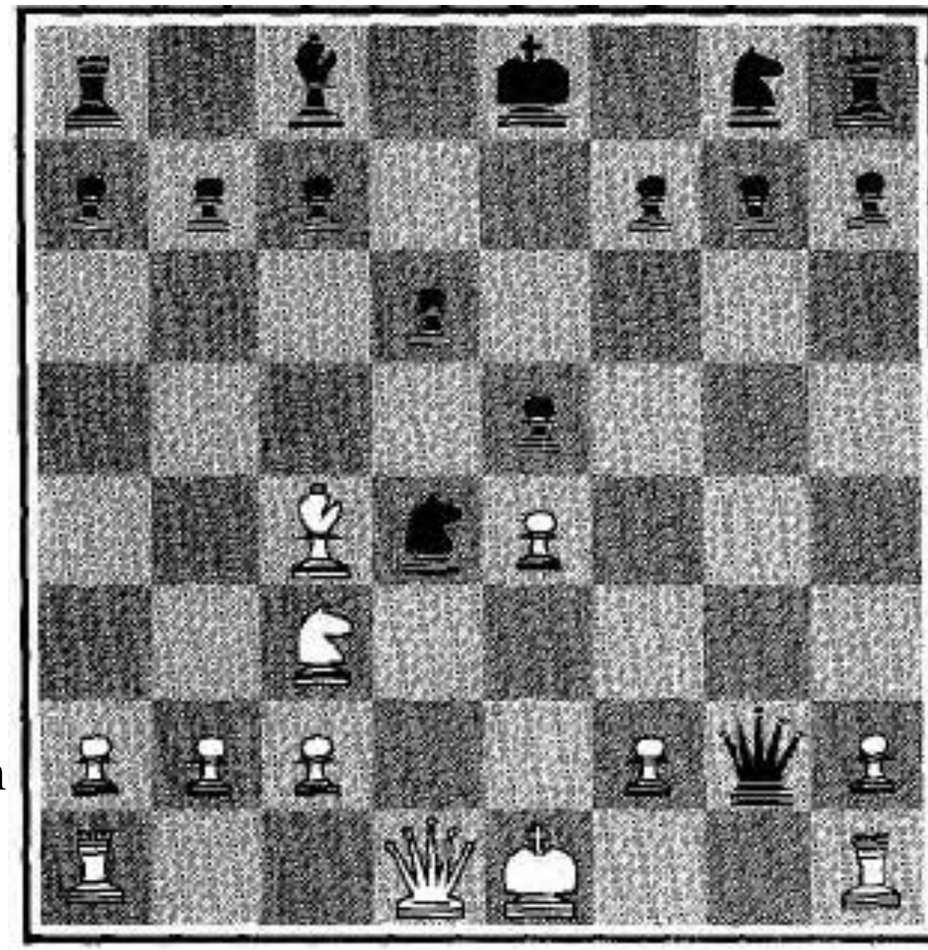
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

$$(v_{ij}) = 4(i-1) + j - 1$$

BIỂU DIỄN TRONG MÁY TÍNH

Dùng xâu ký hiệu,
ví dụ: bàn cờ Châu Âu

"T, XD, x , TgD, x , VD , x , MD, XD ,
ToD,ToD,ToD, x , x ,ToD,ToD,ToD,
x , x , x ,ToD, x , x , x , x ,
x , x , x , x ,ToD, x , x , x ,
x , x ,TgT, MD ,ToT, x , x , x ,
x , x , MT, x , x , x , x , x ,
ToT,ToT ,ToT, x , x ,ToT, HD,ToT,
XT , x , x , HT , VT, x , x , XT ."



x: ô trống, T: quân trắng đến lượt đi
XD: xe đen, TgD: tượng đen, VD: vua đen
MD: mã đen, ToD: tốt đen, HD: hậu đen
TgT: tượng trắng, ToT: tốt trắng,
MT: mã trắng, XT:xe trắng,
HT: hậu trắng, VT: vua trắng

BÀI TOÁN RÓT NƯỚC

- $N = \{(i,j) \mid 0 \le i \le m,\ 0 \le j \le n\}$, $N_0 = \{(0,0)\}$

  Đích = $\{(k,j), (i,k) \mid 0 \le i \le m,\ 0 \le j \le n\}$

  $A = \{(i,j) \rightarrow (m,j),\ (i,j) \rightarrow (i,n),\ (i,j) \rightarrow (0,j),\ (i,j) \rightarrow (i,0),$

  $(i,j) \rightarrow (0,i+j)$ với $i+j \le n$; $\rightarrow (i+j-n,n)$ với $i+j>n$,

  $(i,j) \rightarrow (i+j,0)$ với $i+j \le m$; $\rightarrow (m,i+j-m)$ với $i+j>m\}$

  (*khi nào không có lời giải ?*)

- Ví dụ: m=5, n=6, k=4

  $A = \{(i,j) \rightarrow (5,j),\ (i,j) \rightarrow (i,6),\ (i,j) \rightarrow (0,j),\ (i,j) \rightarrow (i,0),$

  $(i,j) \rightarrow (0,i+j)$ với $i+j \le 6$; $\rightarrow (i+j-6,6)$ với $i+j>6$,

  $(i,j) \rightarrow (i+j,0)$ với $i+j \le 5$; $\rightarrow (5,i+j-5)$ với $i+j>5\}$

- $(0,0) \rightarrow (5,0) \rightarrow (0,5) \rightarrow (5,5) \rightarrow (4,6)$

TRÒ CHƠI n^2-1 SỐ

$N = \{V[n][n] \mid 0 \le v_{ij} \le n^2-1$ và

$\forall(i \ne i'$ hoặc $j \ne j')$: $v_{ij} \ne v_{i'j'}\}$

$N_0 \in N$, Đích $= \{V[n][n] \mid v_{ij} = n*(i-1) + j - 1\}$

$A = \{ V \to W \mid$

tại i,j: $v_{ij}=0$ và $j>1$ thì $\{w_{ij}=v_{i,j-1}; w_{i,j-1}=0; w_{kl}=v_{kl}$ còn lại$\}$,

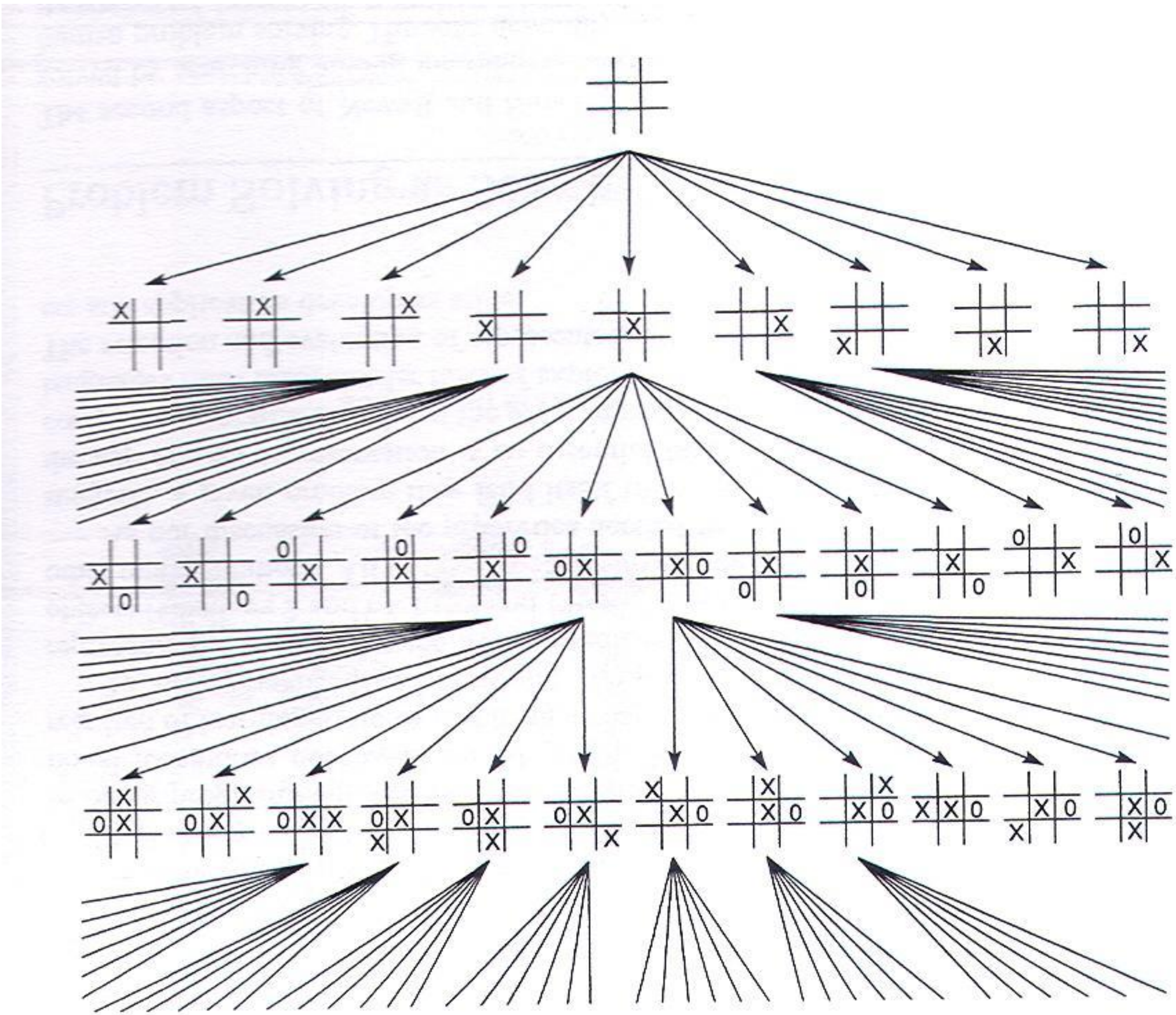
tại i,j: $v_{ij}=0$ và $j<n$ thì $\{w_{ij}=v_{i,j+1}; w_{i,j+1}=0; w_{kl}=v_{kl}$ còn lại$\}$,

tại i,j: $v_{ij}=0$ và $i>1$ thì $\{w_{ij}=v_{i-1,j}; w_{i-1,j}=0; w_{kl}=v_{kl}$ còn lại$\}$,

tại i,j: $v_{ij}=0$ và $i<n$ thì $\{w_{ij}=v_{i+1,j}; w_{i+1,j}=0; w_{kl}=v_{kl}$ còn lại$\}$

$\}$

KGTT của Trò Chơi Tic-Tac-Toe



BÀI TOÁN THÁP HÀ NỘI

- $N = \{(i,j,k) \mid 1 \le i,j,k \le 3$, với i là vị trí của đĩa lớn, j là vị trí của đĩa nhỡ, k là vị trí của đĩa bé$\}$

  $N_0 = \{(1,1,1)\}$, $Đích = \{(3,3,3)\}$

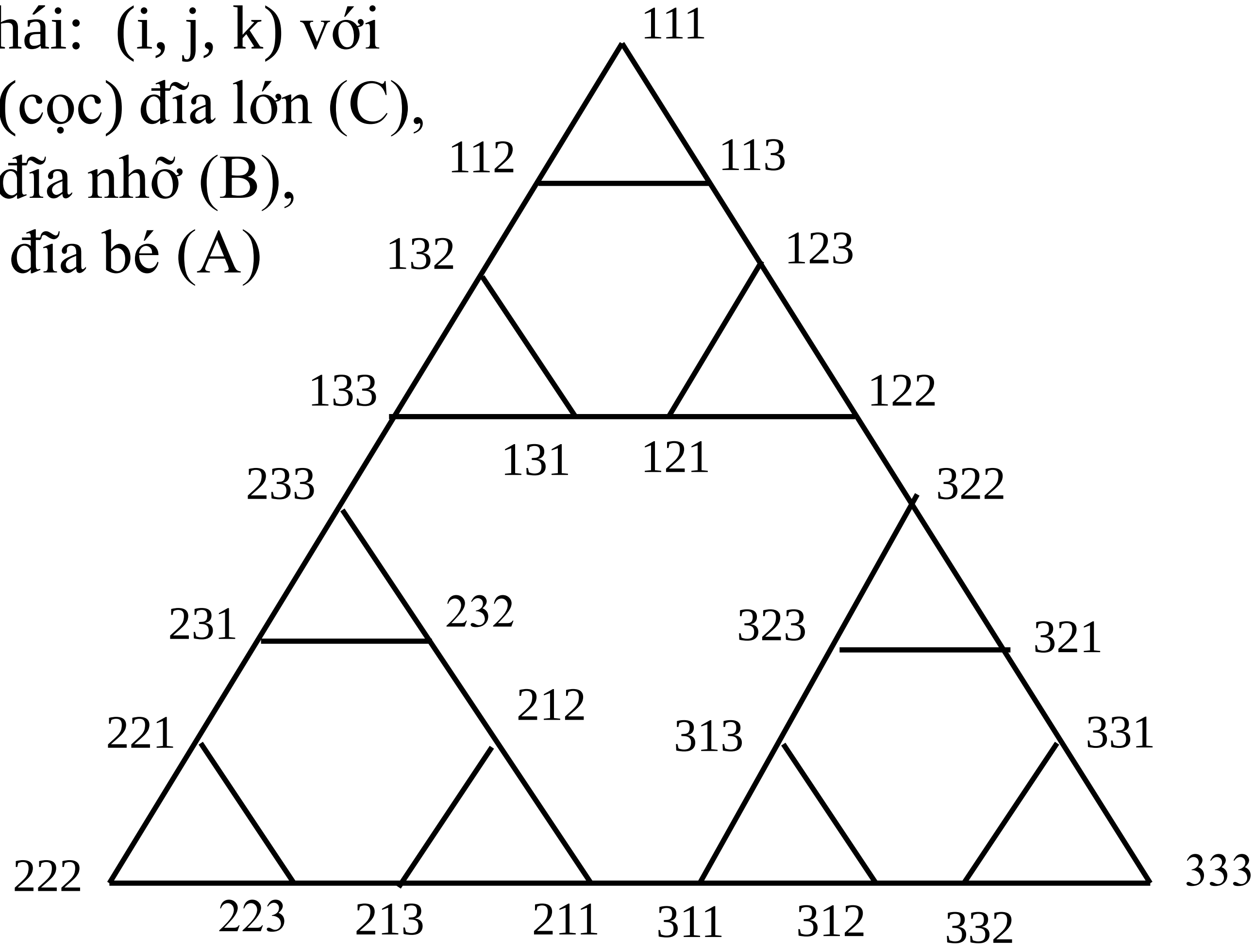
- $A = \{(i,j,k) \rightarrow (i,j,k')$, với $1 \le i,j,k,k' \le 3$ và $k' \ne k$,

  $(i,j,k) \rightarrow (i,j',k)$, với $1 \le i,j,k,j' \le 3$ và $j' \ne j$ và $j \ne k$ và $j' \ne k$,

  $(i,j,k) \rightarrow (i',j,k)$, với $1 \le i,j,k,i' \le 3$ và $i' \ne i$ và $i \ne j$ và $i' \ne j$ và $i \ne k$ và $i' \ne k$ $\}$

Không gian trạng thái của bài toán Tháp Hà Nội

Trạng thái: (i, j, k) với
i: vị trí (cọc) đĩa lớn (C),
j: vị trí đĩa nhỡ (B),
k: vị trí đĩa bé (A)



QUI VỀ CÁC BÀI TOÁN CON

Tháp Hà Nội n = 3

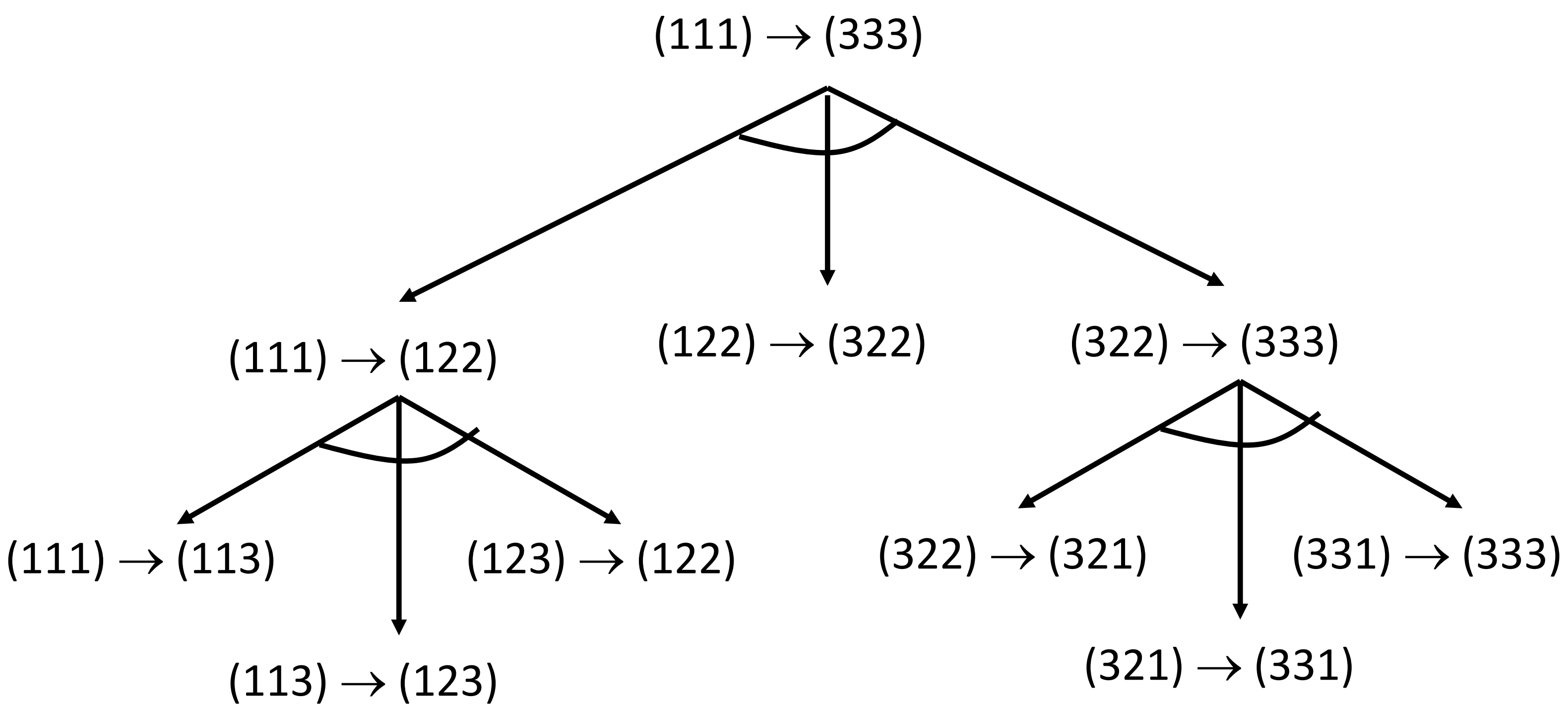
Bài toán đầu (111) → (333) được qui về 3 bài toán con:

BT1. (111) → (122): chuyển 2 đĩa AB từ cọc 1 sang cọc 2

BT2. (122) → (322): chuyển đĩa C từ cọc 1 sang cọc 3

BT3. (322) → (333): chuyển 2 đĩa AB từ cọc 2 sang cọc 3

BT2 giải được ngay, BT1 và BT3 tiếp tục phân rã



3.3. TÌM KIẾM LỜI GIẢI TRÊN KHÔNG GIAN BÀI TOÁN

Phát biểu bài toán:

- Cho trạng thái đầu s_0, cho tập trạng thái đích ĐICH,
- Tìm dãy trạng thái $s_0, s_1, \ldots, s_n$ sao cho: $s_n \in$ ĐICH và $\forall i$: $s_i \rightarrow s_{i+1}$ nhờ áp dụng toán tử biến đổi
- Tìm dãy toán tử $O_1, \ldots, O_{n-1}, O_n$ sao cho: $O_n(O_{n-1}(\ldots O_1(s_0)..)) = s_n \in$ ĐICH
- Tìm dãy sản xuất $p_1, \ldots, p_n$ sao cho $s_0 \Rightarrow^{p1} s_1 \Rightarrow \ldots \Rightarrow^{pn} s_n \in$ ĐICH

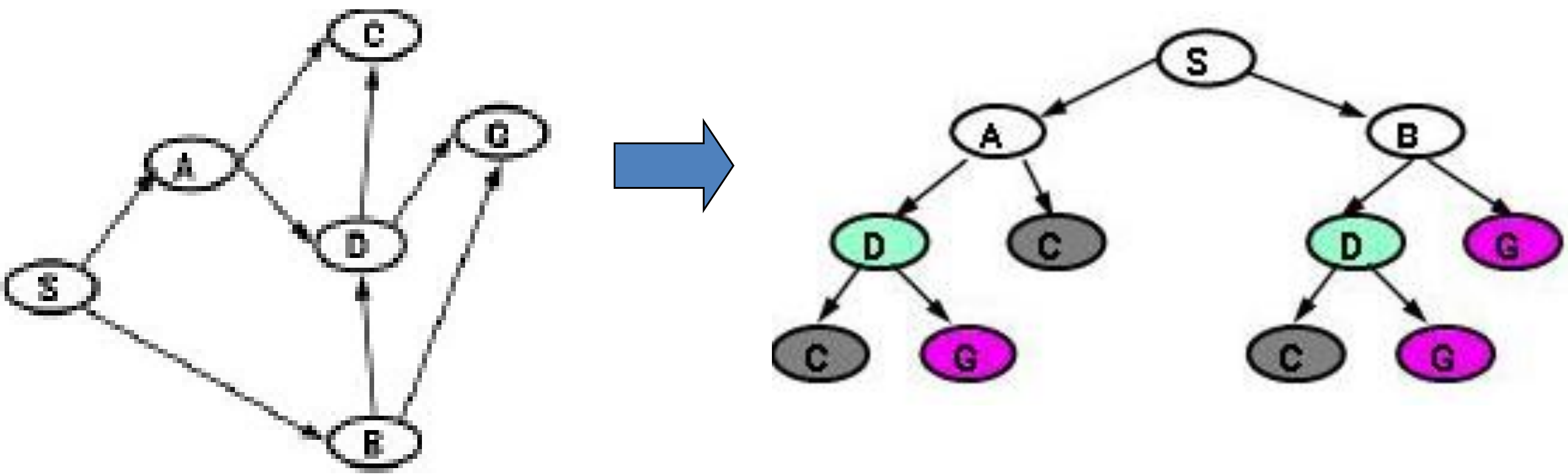
BIỂU DIỄN BẰNG ĐỒ THỊ

Đồ thị G là cặp G = (N,A) với N - tập các nút, A - tập các cung và với $\forall n \in N$: $\Gamma(n) = \{m \in N \mid (n,m) \in A\}$

KGTT	Đồ thị
• Trạng thái (đầu, đích) S, ababb	• nút (đầu, đích)
• Toán tử (sản xuất) $S \rightarrow Sa$	• cung
• Dãy trạng thái liên tiếp	• đường đi
• Dãy toán tử $S \rightarrow Sa \rightarrow aBa$	
• Bài toán P_1, P_2	• Tìm đường đi trên đồ thị từ đỉnh đầu n_0 (tương ứng với s_0) tới đỉnh ĐICH

CHUYỂN VỀ BIỂU DIỄN CÂY

- Cây là đồ thị có hướng không có chu trình và các nút có <= 1 nút cha.
- Chuyển TK trên đồ thị về TK trên cây:
  - thay các liên kết không định hướng bằng 2 liên kết có hướng
  - tránh các vòng lặp trên đường (sử dụng biến tổng thể để lưu vết các nút đã thăm)



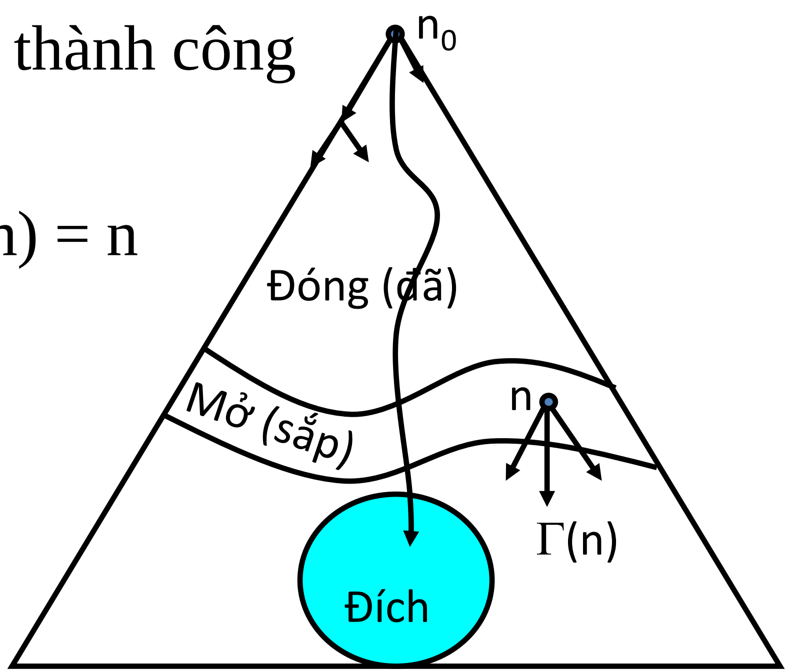
CÁC ĐẶC TÍNH TÌM KIẾM

- Tính đầy đủ
  - Khi bài toán có lời giải thì giải thuật tìm kiếm có thể tìm thấy lời giải không?
- Thời gian
  - Thời gian cần thiết để tìm thấy lời giải
- Không gian
  - Dung lượng nhớ cần thiết để tìm thấy lời giải
- Sự tối ưu
  - Khi có hàm giá, giải thuật có đảm bảo tìm được lời giải tối ưu không?

A/ THUẬT TOÁN TÌM KIẾM CƠ BẢN

Xây dựng tập Mở - tập các đỉnh sắp duyệt; Đóng - tập các đỉnh đã duyệt; n_0 - trạng thái đầu

1. Mở = $\{n_0\}$; Đóng = $\varnothing$
2. Chọn $n \in$ Mở:
   Đóng = Đóng $\cup$ {n}
   Mở = Mở $\cup$ $\Gamma(n)$ // $\Gamma(n)$: tập các nút con của n
3. Lặp (2) đến khi gặp $n^* \in$ Đích $\Rightarrow$ thành công



Đường đi p:

Với mỗi $m \in \Gamma(n)$, thực hiện: cha(m) = n

$p' = n^*, cha(n^*), cha^2(n^*),\ldots,n_0,$

$p = inverse(p')$

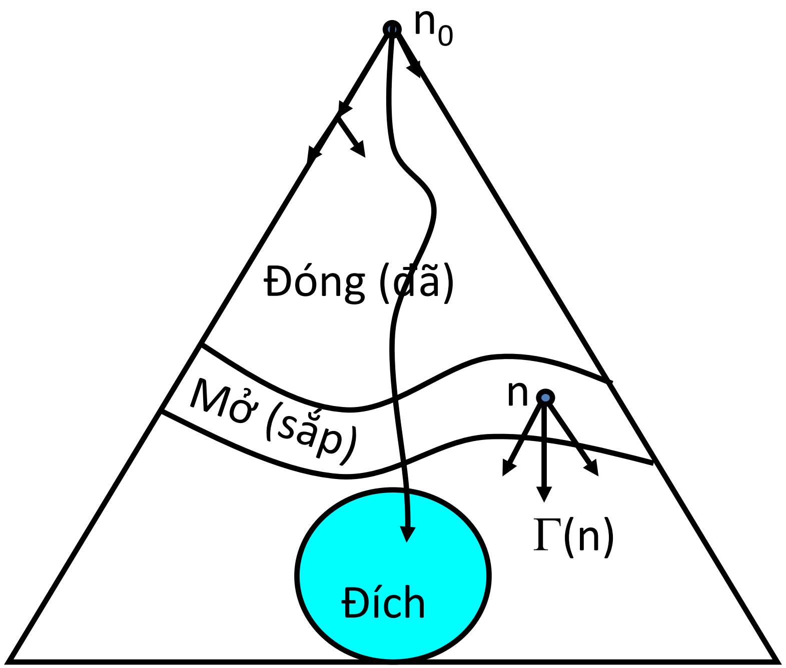
Các quyết định quan trọng:

- Lấy $n \in$ Mở
- Bổ sung $\Gamma(n)$ vào Mở

CÁC CHIẾN LƯỢC TÌM KIẾM

Các quyết định quan trọng:

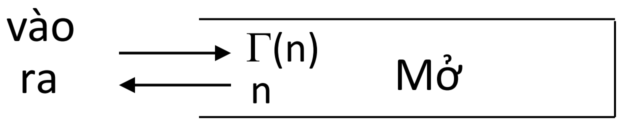
- Lấy n ∈ Mở
- Bổ sung Γ(n) vào Mở



Tìm kiếm sâu (Depth-first):

Vào sau ra trước

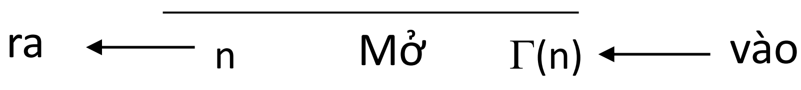
(LIFO – Last In First Out)



Tìm kiếm rộng (Breadth-first):

Vào trước ra trước

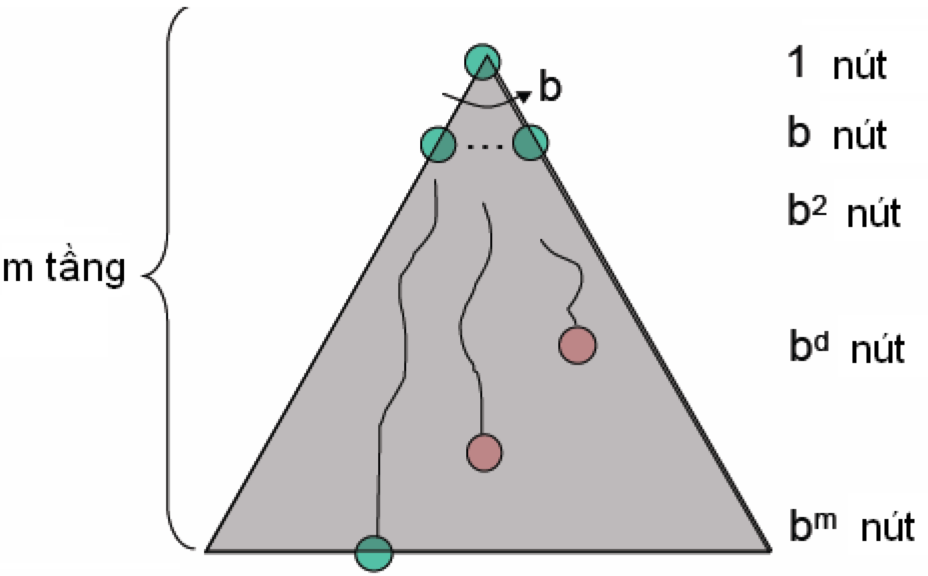
(FIFO – First In First Out)



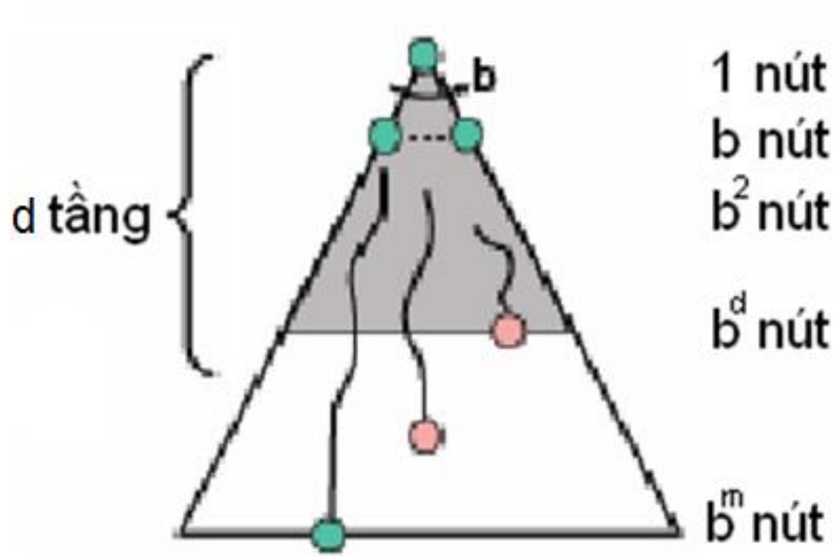
Tìm kiếm cực tiểu (Uniform-cost):

Lấy phần tử có giá nhỏ nhất dựa trên hàm giá

Tìm kiếm sâu



Tìm kiếm rộng



Thuật toán	Đầy đủ	Tối ưu	Thời gian	Không gian
TKS	không	không	$O(b^m)$	$O(b^m)$
TKR	có	không	$O(b^{d+1})$	$O(b^{d+1})$

$= 1 + b + b^2 + \ldots + b^d + b(b^d-1) = O(b^{d+1})$

VÍ DỤ - BÀI TOÁN RÓT NƯỚC

- Mở = (0,0), Đóng = ∅, n=(0,0)
- n=(0,0), Γ(n)= {(5,0), (0,6)}, Đóng = {(0,0)}
  TKS: Mở = {(5,0), (0,6)}, TKR: Mở = {(5,0), (0,6)}
- n=(5,0), Γ(n)= {(0,5), (5,6)}, Đóng = {(0,0), (5,0)}
  TKS: Mở = {(0,5), (5,6), (0,6)}, TKR: Mở = {(0,6), (0,5), (5,6)}

TKS

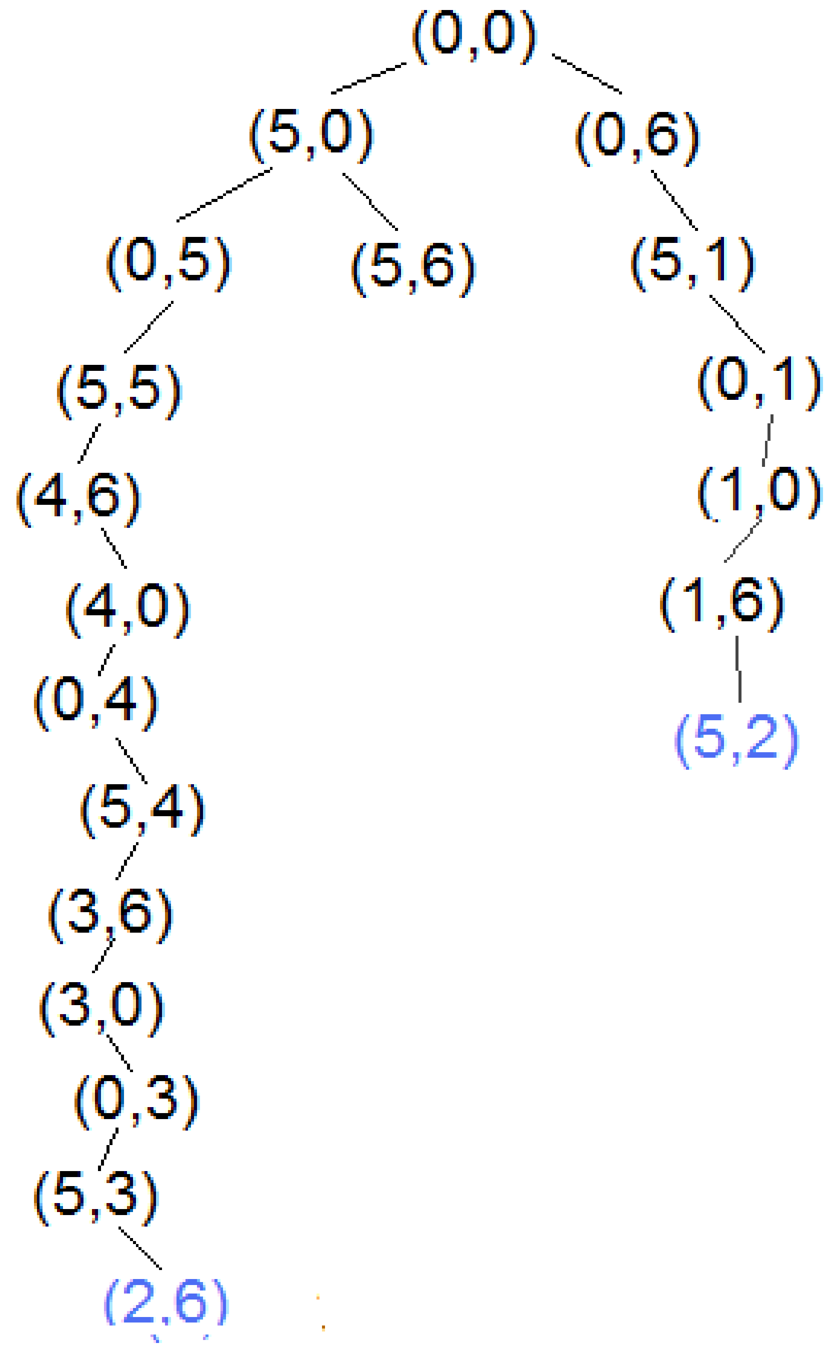
- n=(0,5), Γ(n)={(5,5)}, Mở ={(5,5)..}
- n=(5,5), Γ(n)={(4,6)}, Mở ={(4,6)..}
- n=(4,6), Γ(n)={(4,0)}, Mở ={(4,0)..}
- n=(4,0), Γ(n)={(0,4)}, Mở ={(0,4)..}
- n=(0,4), Γ(n)={(5,4)}, Mở ={(5,4)..}
- n=(5,4), Γ(n)={(3,6)}, Mở ={(3,6)..}
- n=(3,6), Γ(n)={(3,0)}, Mở ={(3,0)..}
- n=(3,0), Γ(n)={(0,3)}, Mở ={(0,3)..}
- n=(0,3), Γ(n)={(5,3)}, Mở ={(5,3)..}
- n=(5,3), Γ(n)={(2,6)}

TKR

- n=(0,6), Γ(n)={(5,1)}, Mở ={(0,5)..}
- n=(0,5), Γ(n)={(5,5)}, Mở ={(5,6)..}
- n=(5,6), Γ(n)=∅, Mở ={(5,1)..}
- n=(5,1), Γ(n)={(0,1)}, Mở ={(5,5)..}
- n=(5,5), Γ(n)={(4,6)}, Mở ={(0,1)..}
- n=(0,1), Γ(n)={(1,0)}, Mở ={(4,6)..}
- n=(4,6), Γ(n)={(4,0)}, Mở ={(1,0)..}
- n=(1,0), Γ(n)={(1,6)}, Mở ={(4,0)..}
- n=(4,0), Γ(n)={(0,4)}, Mở ={(1,6)..}
- n=(1,6), Γ(n)={(5,2)}

TKS hay TKR

- Có cần thiết tìm một *đường đi ngắn nhất* đến mục tiêu hay không?
- Sự *phân nhánh* của không gian trạng thái
- Tài nguyên về *không gian* và *thời gian* sẵn có
- *Khoảng cách trung bình* của đường dẫn đến trạng thái mục tiêu.
- Yêu cầu đưa ra *tất cả các lời giải* hay chỉ là lời giải tìm được đầu tiên.



TÌM KIẾM SÂU DẦN

- TKS có thể cho kết quả nhưng đường đi không phải là ngắn nhất
  Tuy có thể tồn tại đường đi đến Đích nhưng TKS có thể không dừng.

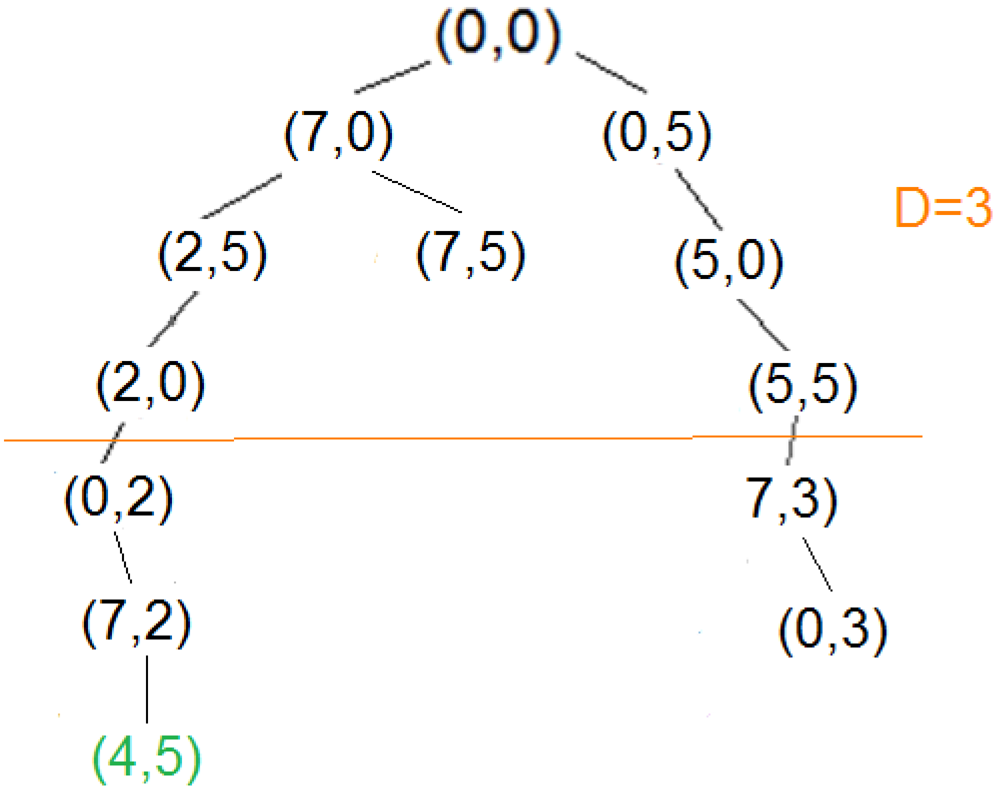
⇒ chọn ngưỡng sâu D, mỗi đỉnh được gán một độ sâu d(n)

Lấy n ∈ Mở, nếu d(n) < D, như TKS,
nếu d(n) = D, như TKR,
nếu d(n) > D, tăng ngưỡng sâu thêm D

⇒ Tìm kiếm sâu dần: tăng dần D !

Thuật toán	Đầy đủ	Tối ưu	Thời gian	Không gian
TKSD	có	không	$O(b^d)$	$O(b^d)$

- Bài toán rót nước (7, 5) $\Rightarrow$ 4
- TKR, TKS, TKSD ?



TÌM KIẾM CỰC TIỂU

$c(n_i, n_j)$: chi phí đi từ n_i đến n_j

Xét $p = n_0, n_1, \ldots, n_k$

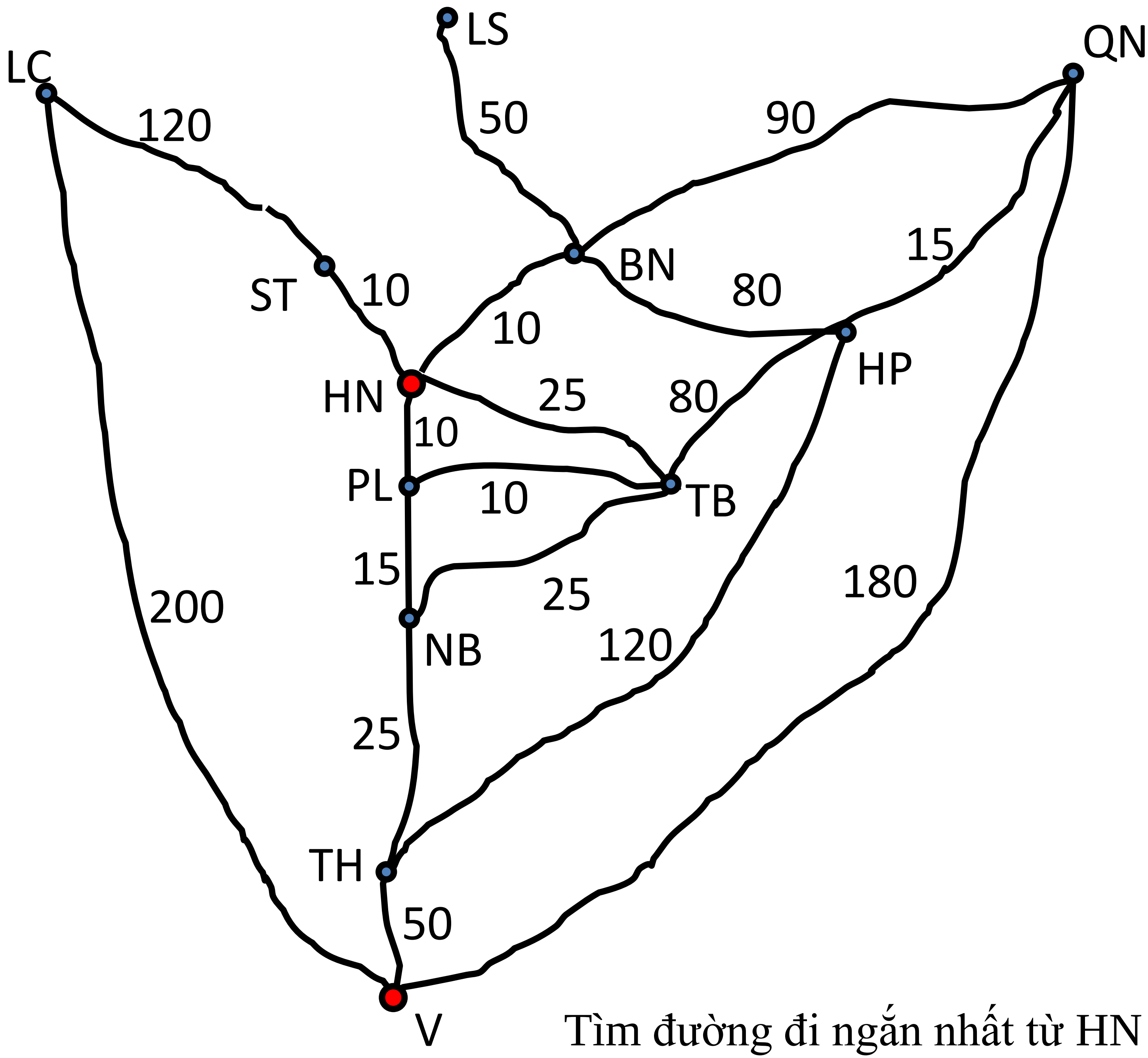
Hàm đánh giá

$c(p) = c(n_0,n_1) + c(n_1,n_2) + \ldots + c(n_{k-1},n_k)$

Lấy $n \in$ Mở: $g(n) = c(p(n_0,n)) \rightarrow \min$

Nếu $\forall c(n_i,n_j) > \varepsilon$, C^* là chi phí của lời giải tối ưu

Thuật toán	Đầy đủ	Tối ưu	Thời gian	Không gian
TKCT	có	có	$O(b^{ceiling(C^*/\varepsilon)})$	$O(b^{ceiling(C^*/\varepsilon)})$



n	h(n)
HN	100
ST	110
LC	200
BN	110
LS	160
HP	170
QN	180
TB	100
PL	90
NB	75
TH	50
V	0

Tìm đường đi ngắn nhất từ HN đến V

TÌM KIẾM CỰC TIỂU VỚI TRI THỨC BỔ SUNG (A*)

$c(n_i, n_j)$ = chi phí đi từ n_i đến n_j

$g(n)$ = chi phí thực tế đường đi từ n_0 đến n

$h(n)$ = chi phí ước lượng đường đi từ n đến đích, (do chuyên gia cung cấp !)

- $h(n)$ chấp nhận được nếu với $\forall n,\ 0 \le h(n) \le h^*(n)$, trong đó $h^*(n)$ là chi phí thực để tới trạng thái đích từ n.
- $h(n)$ càng sát với $h^*(n)$ thì thuật toán càng mạnh

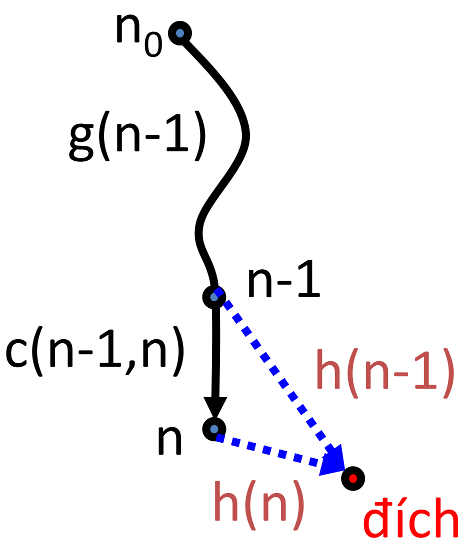
$f(n) = g(n) + h(n)$

$f(n-1) = g(n-1) + h(n-1)$

$g(n) = g(n-1) + c(n-1,n)$

$f(n) = g(n-1) + c(n-1,n) + h(n)$

$= f(n-1) – h(n-1) + c(n-1,n) + h(n)$



Lấy $n \in$ Mở: $f(n) \to \min$!

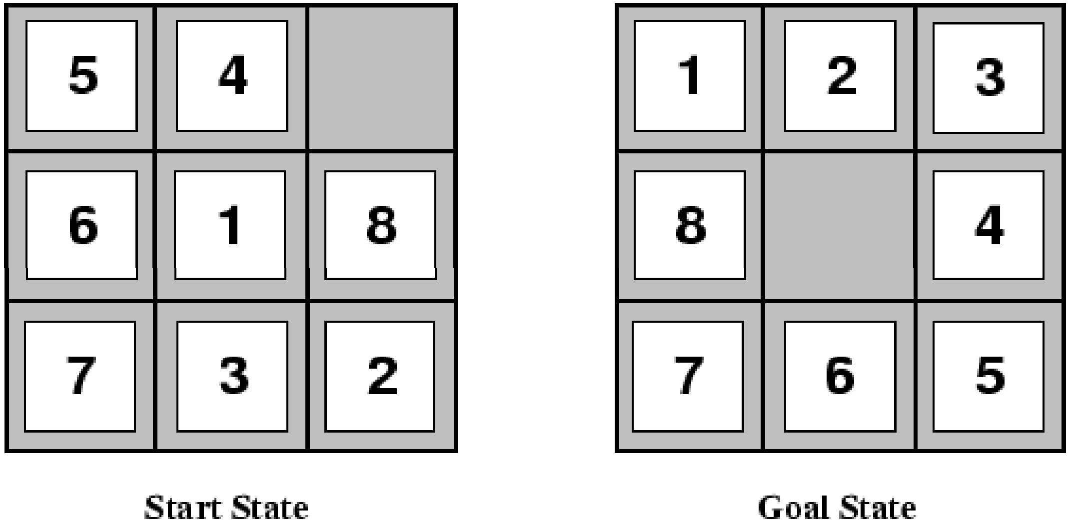
B/ TÌM KIẾM SỬ DỤNG HEURISTIC

- (heuristic ~ dựa trên kinh nghiệm để GQVĐ, không đảm bảo tối ưu)
- Nguyên lý: vét cạn thông minh, tham lam, thứ tự, hàm heuristic
- Hàm Heuristic: $f(n) = g(n) + h(n)$

  g: hướng lợi ích

  h: hướng đích
- Tại nút (n-1) chọn nút tiếp theo $n \in$ Mở dựa theo $f(n)$

VÍ DỤ - TRÒ CHƠI 8 SỐ



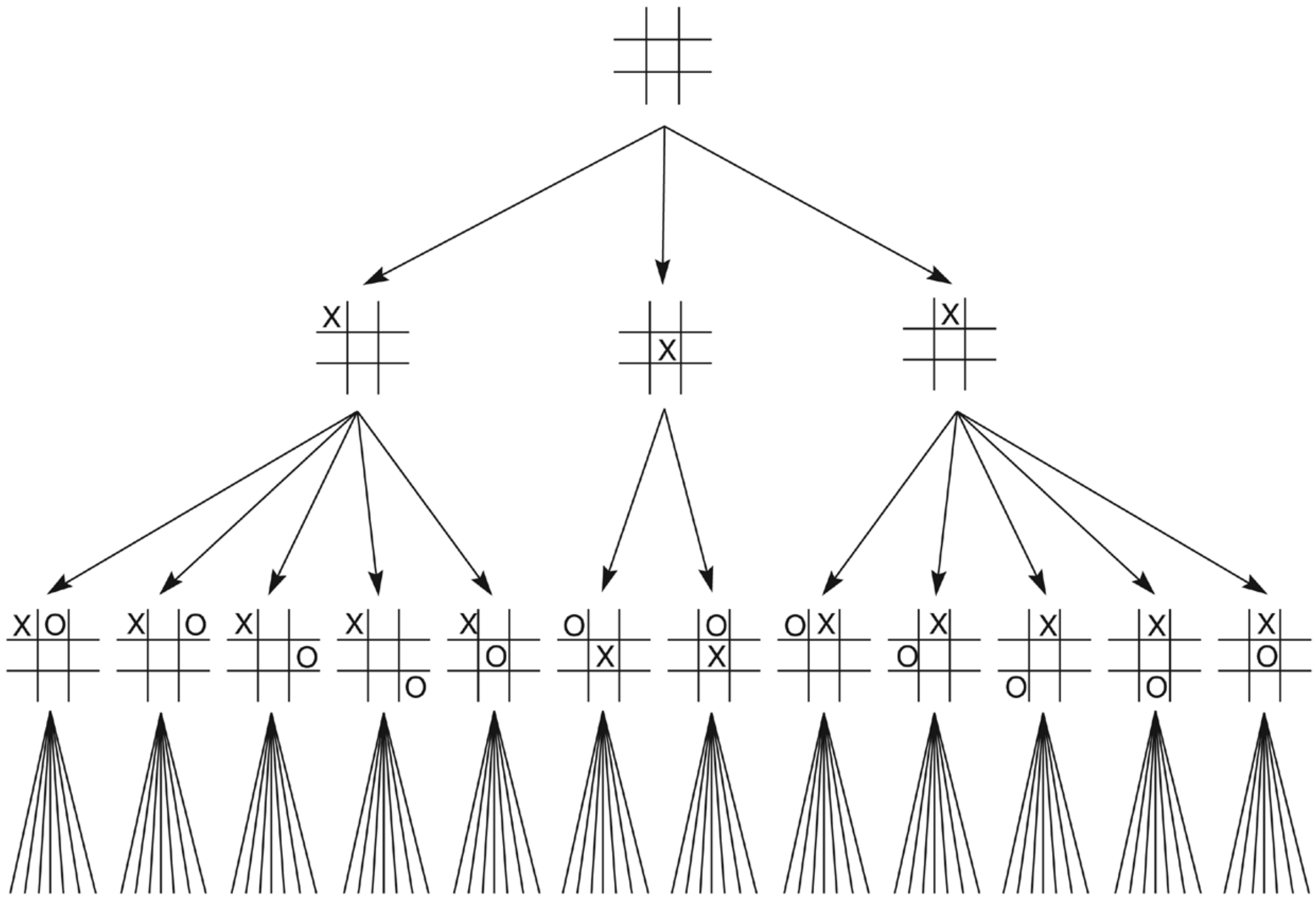
Ví dụ về heuristic

- Số viên sai vị trí (H1)
- Khoảng cách Manhattan (H2). (Khoảng cách Manhattan giữa (x_1,y_1) và (x_2,y_2) là $|x_1-x_2|+|y_1-y_2|$)

H1(S) = 7

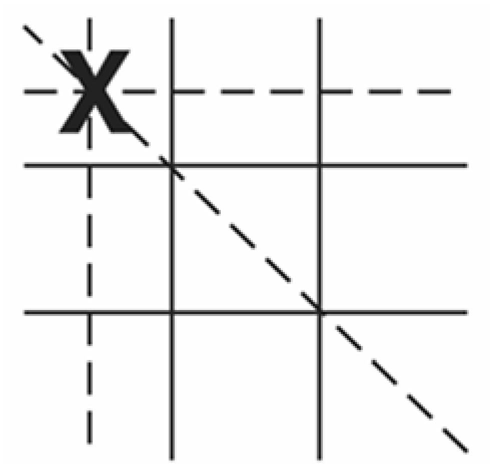
H2(S) = 2+3+3+2+4+2+0+2 = 18

TRÒ CHƠI TIC-TAC-TOE

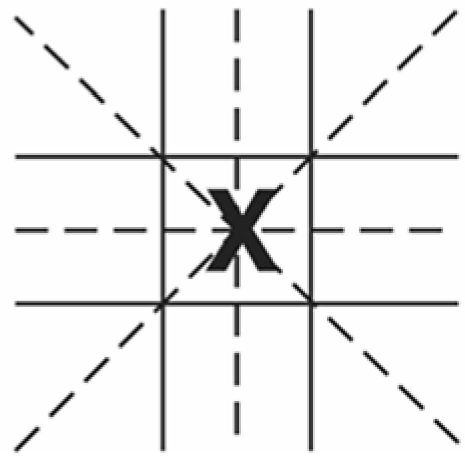


KGTT của tic-tac-toe được thu nhỏ nhờ tính đối xứng của các trạng thái

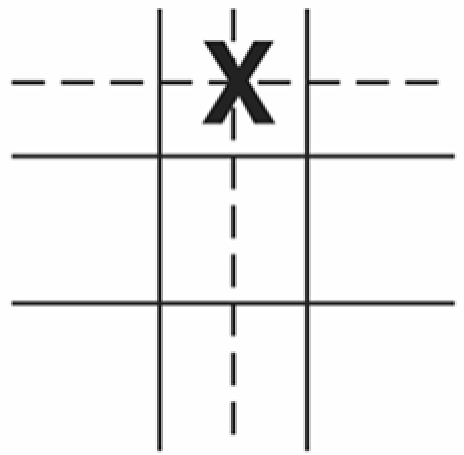
HEURISTIC



Chiếm 3 đường



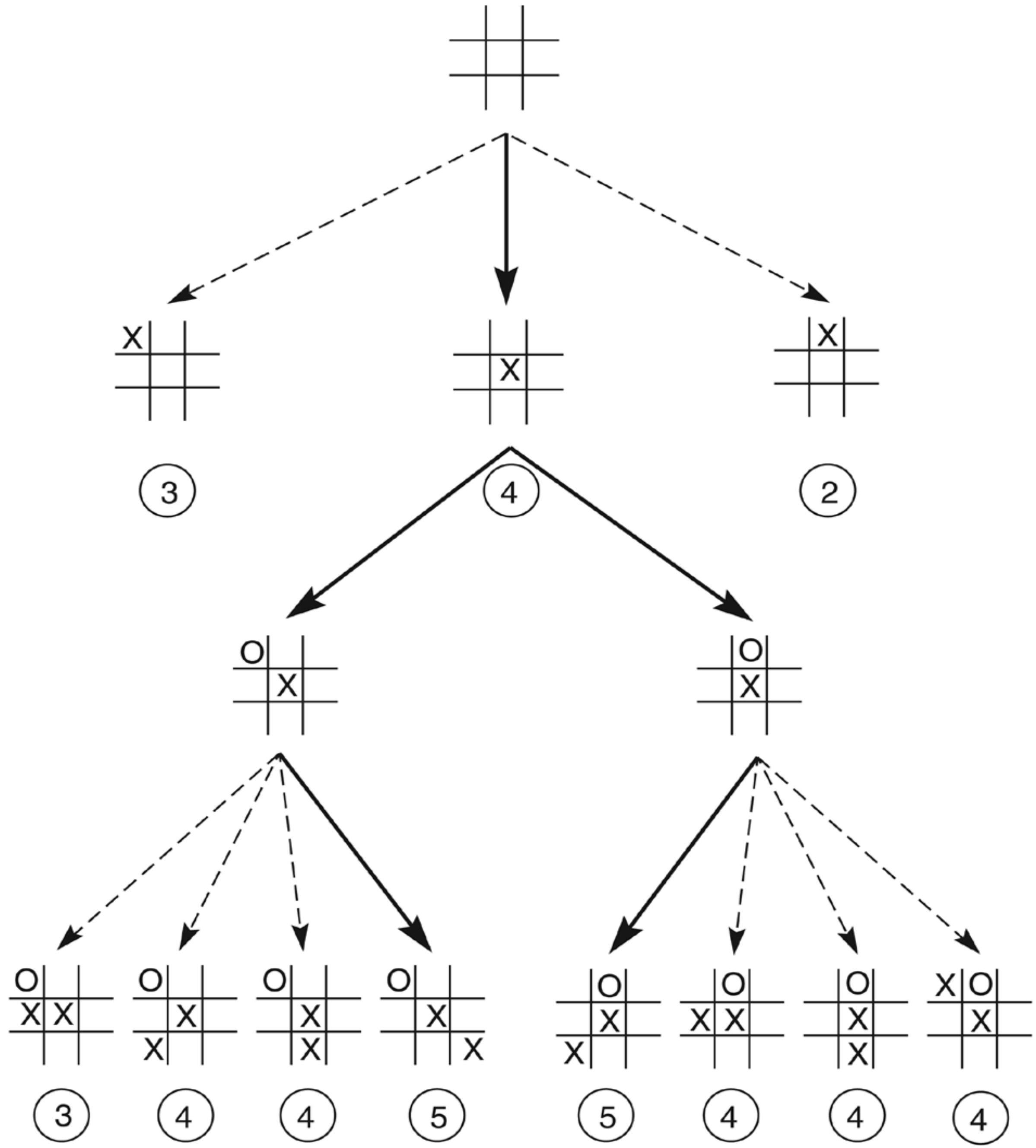
Chiếm 4 đường



Chiếm 2 đường

Heuristic “*Số đường thẳng nhiều nhất*” áp dụng cho các nút con đầu tiên trong tic-tac-toe.

Độ đo heuritic cho “×” ở nước đi thứ nhất và thứ hai



C/ TÌM KIẾM TRÊN ĐỒ THỊ Và/Hoặc

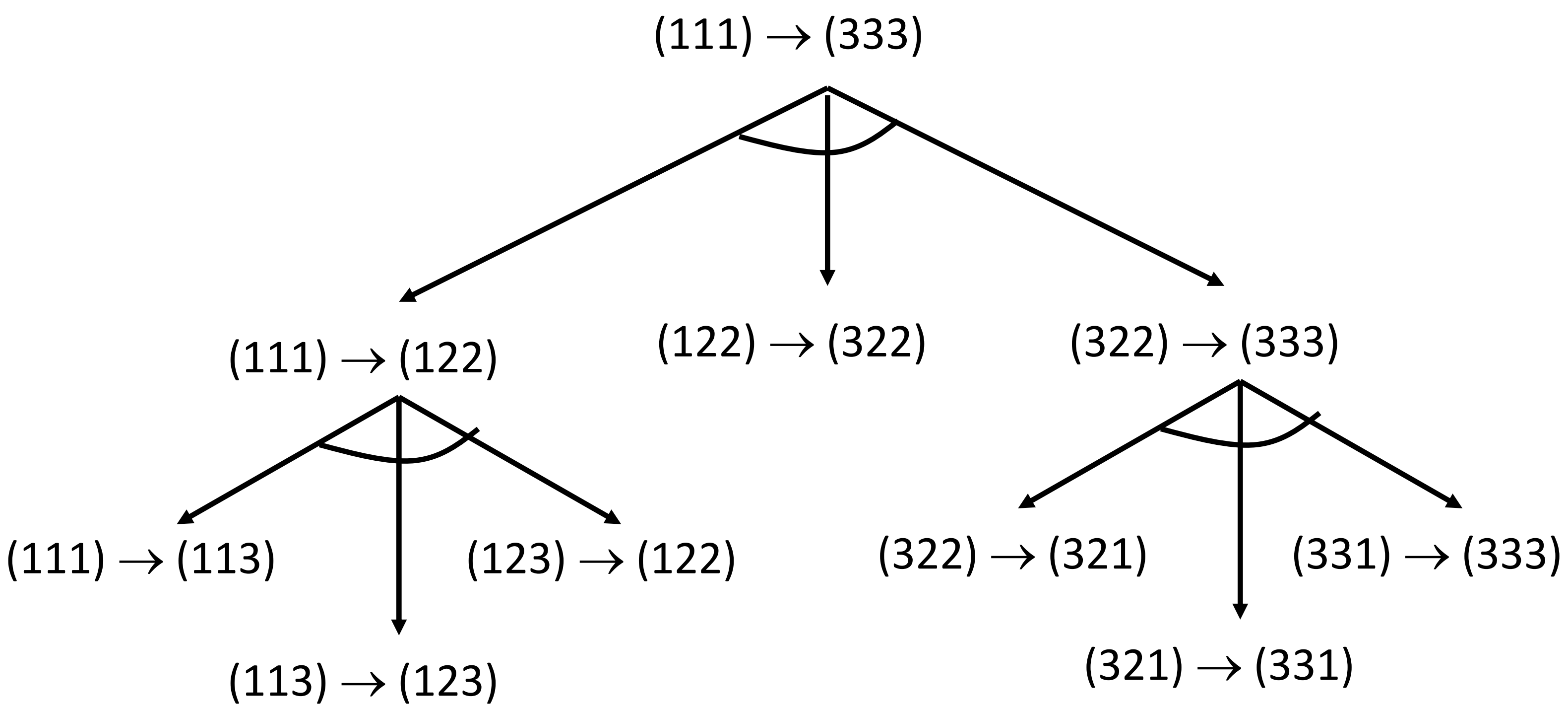
Ví dụ: Bài toán Tháp Hà Nội n = 3, Bài toán đầu (111) → (333) được qui về ba bài toán con:

BT1. (111) → (122): chuyển 2 đĩa AB từ cọc 1 sang cọc 2

BT2. (122) → (322): chuyển đĩa C từ cọc 1 sang cọc 3

BT3. (322) → (333): chuyển 2 đĩa AB từ cọc 2 sang cọc 3

BT2 giải được ngay, BT1 và BT3 tiếp tục phân rã



Qui bài toán về BT con

- Bài toán
- Qui về BT con
  - Phát biểu lại BT

    $P \rightarrow P_1,\ldots,P_n$
  - Phân rã P thành

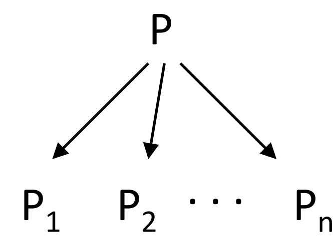
    $P_1,\ldots,P_n$
- BT xuất phát
- BT sơ cấp (nguyên tử) (∃ thuật giải để giải quyết)
- Giải bài toán P

Đồ thị Và/Hoặc

- Đỉnh
- Cung
  - Cung Hoặc $P \rightarrow P_1$

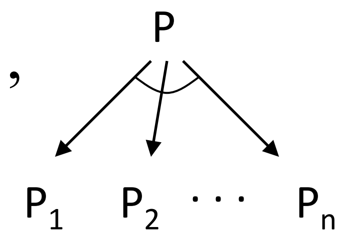
    …

    $P \rightarrow P_n$



  - Cung Và $P \rightarrow P_{1,}, \ldots,$

    $P \rightarrow P_n$

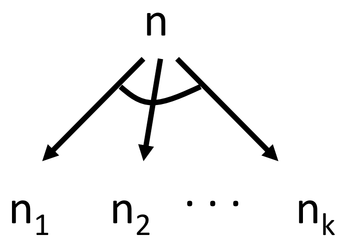


- Đỉnh gốc n_0
- Đỉnh kết thúc
- Xây dựng đồ thị lời giải

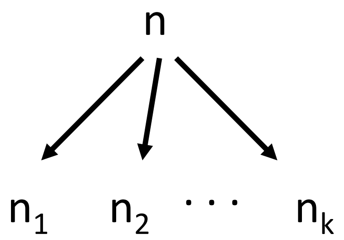
Đỉnh giải được

1. Đỉnh kết thúc ($\Leftrightarrow$ bài toán sơ cấp) giải được
2. Giả sử n có con $n_1, \ldots, n_k$,
   - $n_1, \ldots, n_k \in N_{VÀ}$: n giải được $\Leftrightarrow \forall n_i$ giải được
   - $n_1, \ldots, n_k \in N_{HOẶC}$: n giải được $\Leftrightarrow \exists n_i$ giải được

Đỉnh không giải được



1. n là lá ($\Gamma(n) = \varnothing$), n không kết thúc
2. n có con $n_1, \ldots, n_k$
   - $n_i \in N_{VÀ}$, $n_{kgd} \Leftrightarrow \exists\ n_i$ không giải được
   - $n_i \in N_{HOẶC}$, $n_{kgd} \Leftrightarrow \forall\ n_i$ không giải được



Qui ước: $n \in N$ là một bài toán nào đó

nhan(n) = gd, nếu đỉnh n giải được

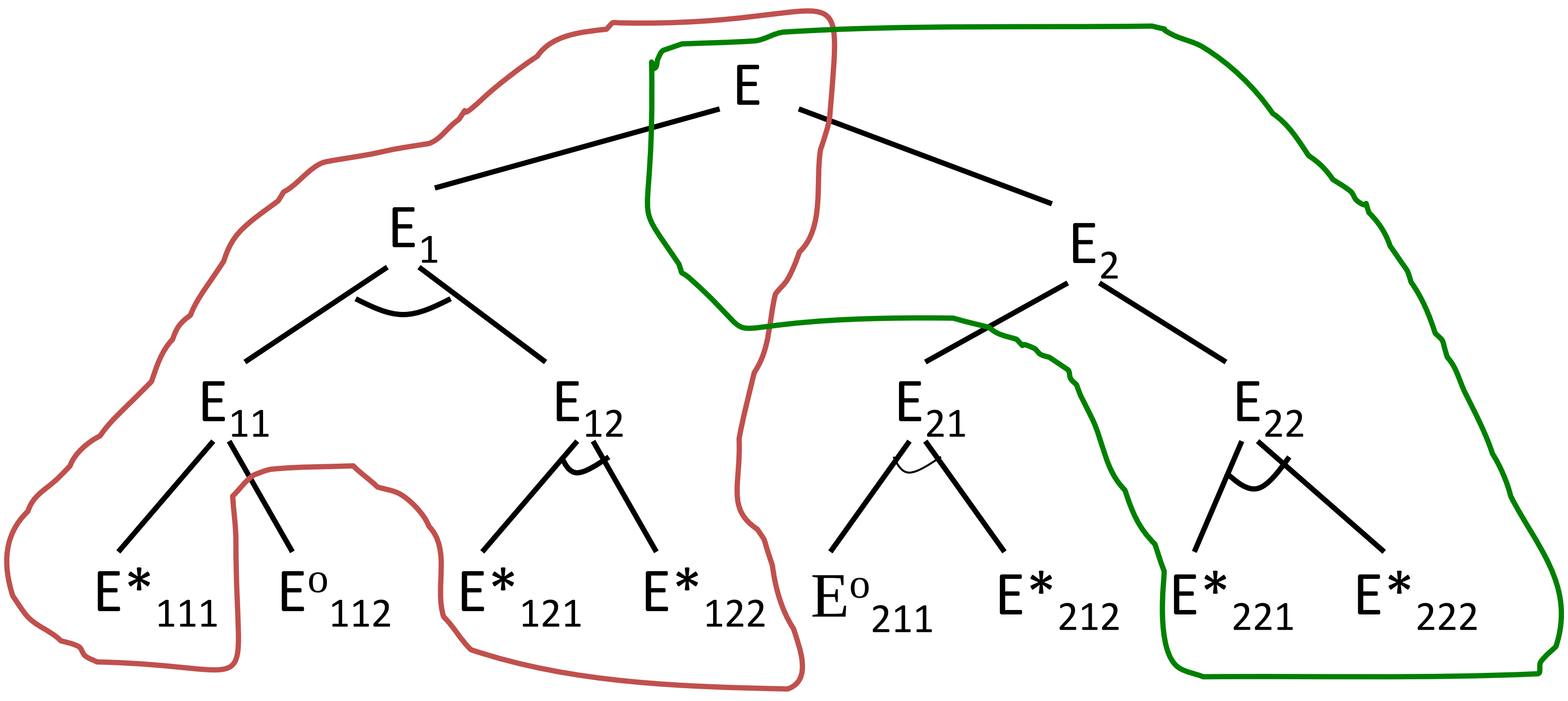
kgd, nếu đỉnh n không giải được

kxd, nếu đỉnh n không xác định

Với bài toán P, cần xác định nhãn(n_0), kéo theo đồ thị lời giải

CÂY LỜI GIẢI T

- là cây con của đồ thị G
  - $n_0 \in T$
  - $\forall$ đỉnh $n \in T$, n giải được



Thuật toán gán nhãn đỉnh giải được

procedure GD(n)

/* n gd hay nhan(n) = true tuỳ thuộc vào thông tin gd của các đỉnh con của n */

$\{_1$ if kt(n) then nhan(n)=true

else $\{_2$

if n có các đỉnh con là đỉnh VÀ $n_1,\ldots,n_k$ then

$\{_3$ gd(n_1);…; gd(n_k);

if (nhan(n_1) and … and nhan(n_k) then nhan(n) = true $\}_3$

if n có các đỉnh con là đỉnh HOẶC $n_1,\ldots,n_k$ then

$\{_4$ gd(n_1);…;gd(n_k);

if (nhan(n_1) or … or nhan(n_k) then nhan(n) = true $\}_4$

$\}_2$

$\}_1$

Thuật toán gán nhãn đỉnh không giải được

procedure KGD(n)

$\{_1$ if not kt(n) then nhan(n)=false

else $\{_2$

if n có các đỉnh con là đỉnh VÀ $n_1,\ldots,n_k$ then

$\{_3$ gd(n_1);…; gd(n_k);

if (not nhan(n_1) or… or not nhan(n_k)) then nhan(n) = false$\}_3$

if n có các đỉnh con là đỉnh HOẶC $n_1,\ldots,n_k$ then

$\{_4$ gd(n_1);…;gd(n_k);

if (not nhan(n_1) and … and not nhan(n_k)) then nhan(n) = false$\}_4$

$\}_2$

$\}_1$

TK RỘNG TRÊN ĐỒ THỊ VÀ/HOẶC

Vào: Cây Và/Hoặc G=(N,A) với đỉnh đầu n_0

Ra: cây lời giải: /* sử dụng 2 danh sách queue MO, DONG*/

TK sâu:
MO là stack

$\{_1$ MO = $\{n_0\}$; DONG = $\varnothing$;

while MO $\neq \varnothing$ do $\{_2$

n $\leftarrow$ get(MO); DONG $\leftarrow$ DONG $\cup$ $\{n\}$;

if $\Gamma(n) \neq \varnothing$ then $\{_3$ MO $\leftarrow$ MO $\cup$ $\Gamma(n)$;

if trong $\Gamma(n)$ có đỉnh m kết thúc then $\{_4$

nhan(m) = true; gd(n_0); // gọi thủ tục gd

if nhan(n_0) then exit('thanh cong')

else Loại khỏi MO các đỉnh có tổ tiên là đỉnh giải được

$\}_4$ $\}_3$

else$\{_5$ nhan(n) = false; kgd(n_0); // gọi thủ tục kgd

if not nhan(n_0) then exit('khong thanh cong')

else Loại khỏi MO các đỉnh có tổ tiên là đỉnh không giải được $\}_5$

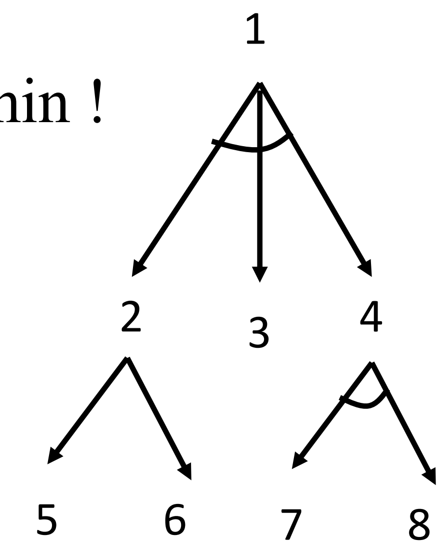
$\}_2$ $\}_1$

Tìm kiếm cực tiểu trên đồ thị Và/Hoặc

Đặt vấn đề: Tìm cây T* để chi phí (giá) C(T*) → min !

Giá của cây T : có thể (tổng hoặc max)

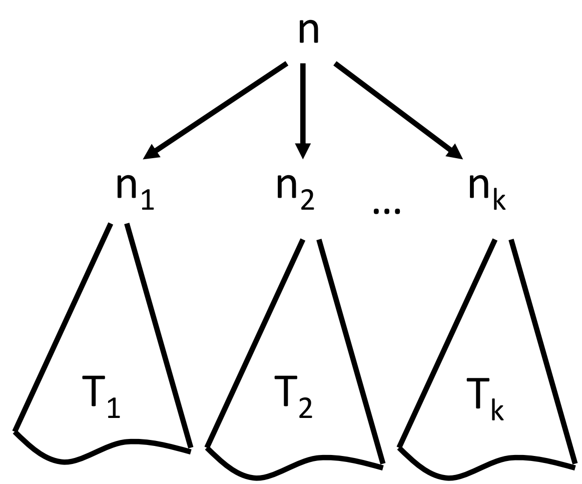
$$C_{\sum}(T) = \sum_{a \in T} c(a) \quad \text{hoặc} \quad C_{\max}(T) = \max_{p: n_0 \to leaves} (c(p))$$



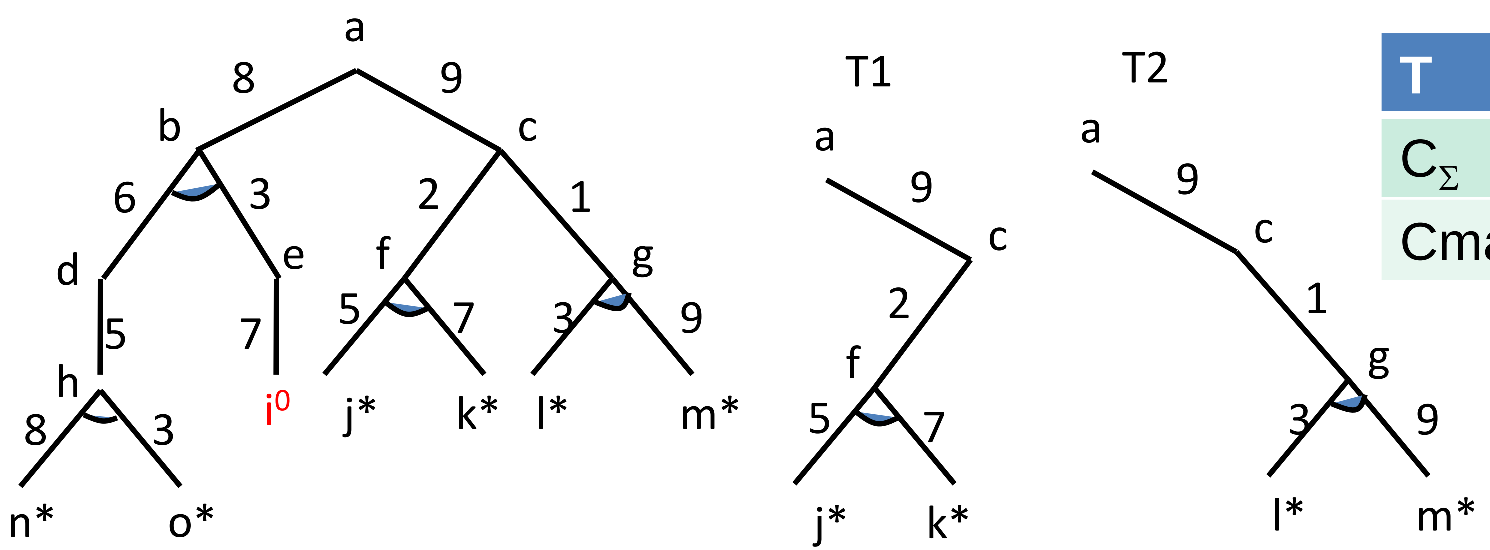
Giá tối ưu để giải bài toán là h(n),

h(n) có các tính chất sau:

- Nếu n là đỉnh kết thúc, h(n) = 0
- Nếu n có con là $n_1, \ldots, n_k$
- $\forall n_i \in N_{HOẶC}$: $h(n) = \min_i [h(n_i) + c(n,n_i)]$
- $\forall n_i \in N_{VÀ}$:
  - Giá tổng: $h(n) = [\sum h(n_i) + \sum c(n,n_i)]$
  - Giá max: $h(n) = \max_i [h(n_i) + c(n,n_i)]$
- Nếu n là đỉnh không giải được, $h(n) = \infty$

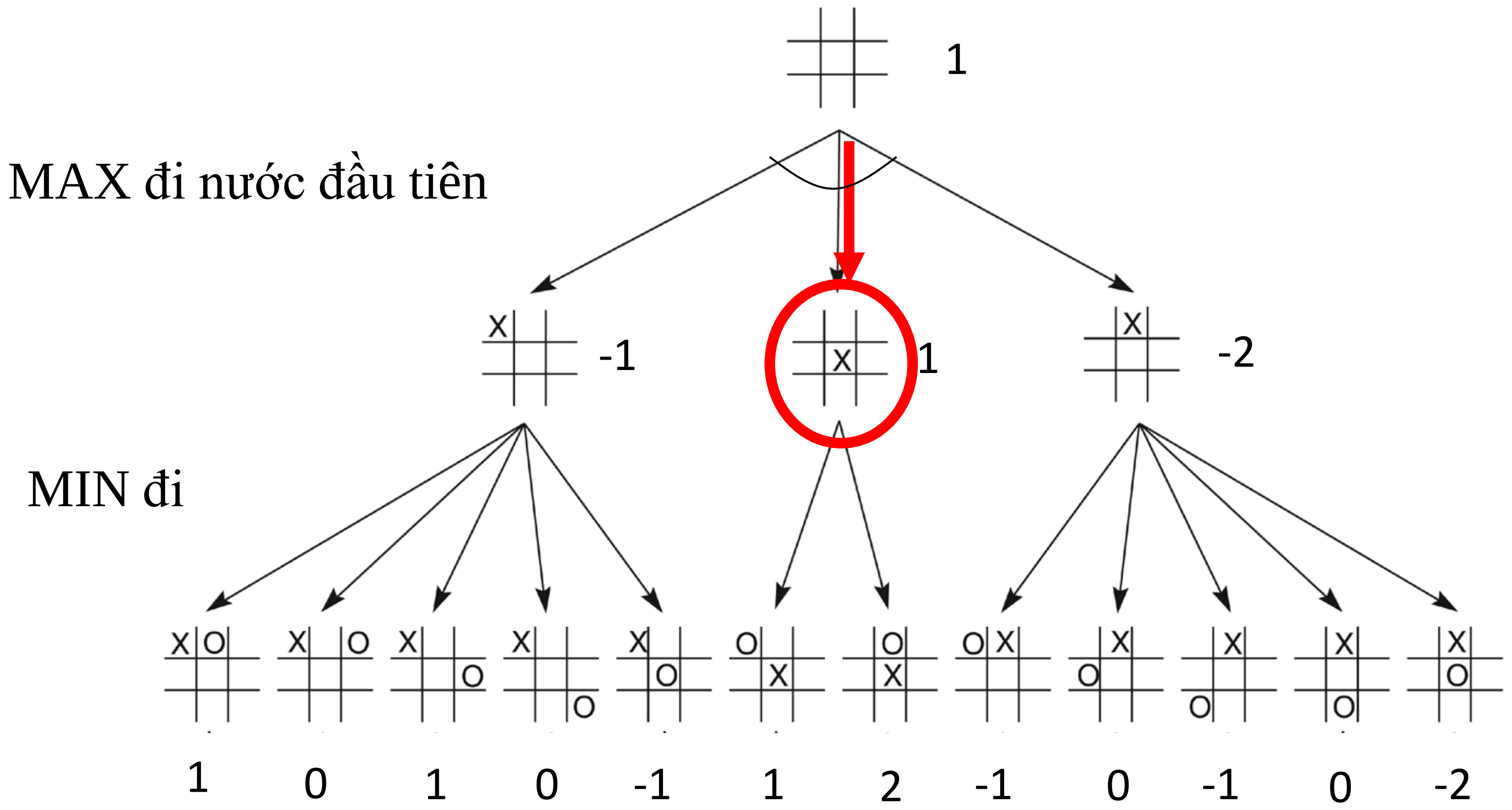


VÍ DỤ



T	T1	T2
C_Σ	23	22
Cmax	18	19

n	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
h_Σ	22	∞	13	16	∞	12	12	11	∞	0	0	0	0	0	0
h_{max}	18	∞	9	13	∞	7	9	8	∞	0	0	0	0	0	0



e(p) = số hàng cột chéo còn mở với ‘x’ - số hàng cột chéo còn mở với ‘o’

D/ TÌM KIẾM CÓ ĐỐI THỦ

Có 2 đối thủ MAX và MIN

- MAX tìm cách làm cực đại một hàm ước lượng nào đó: Chọn nước đi ứng với Giá trị lớn nhất
- MIN tìm cách làm cực tiểu và chọn nước đi ứng với Giá trị nhỏ nhất

Ở mỗi thời điểm:

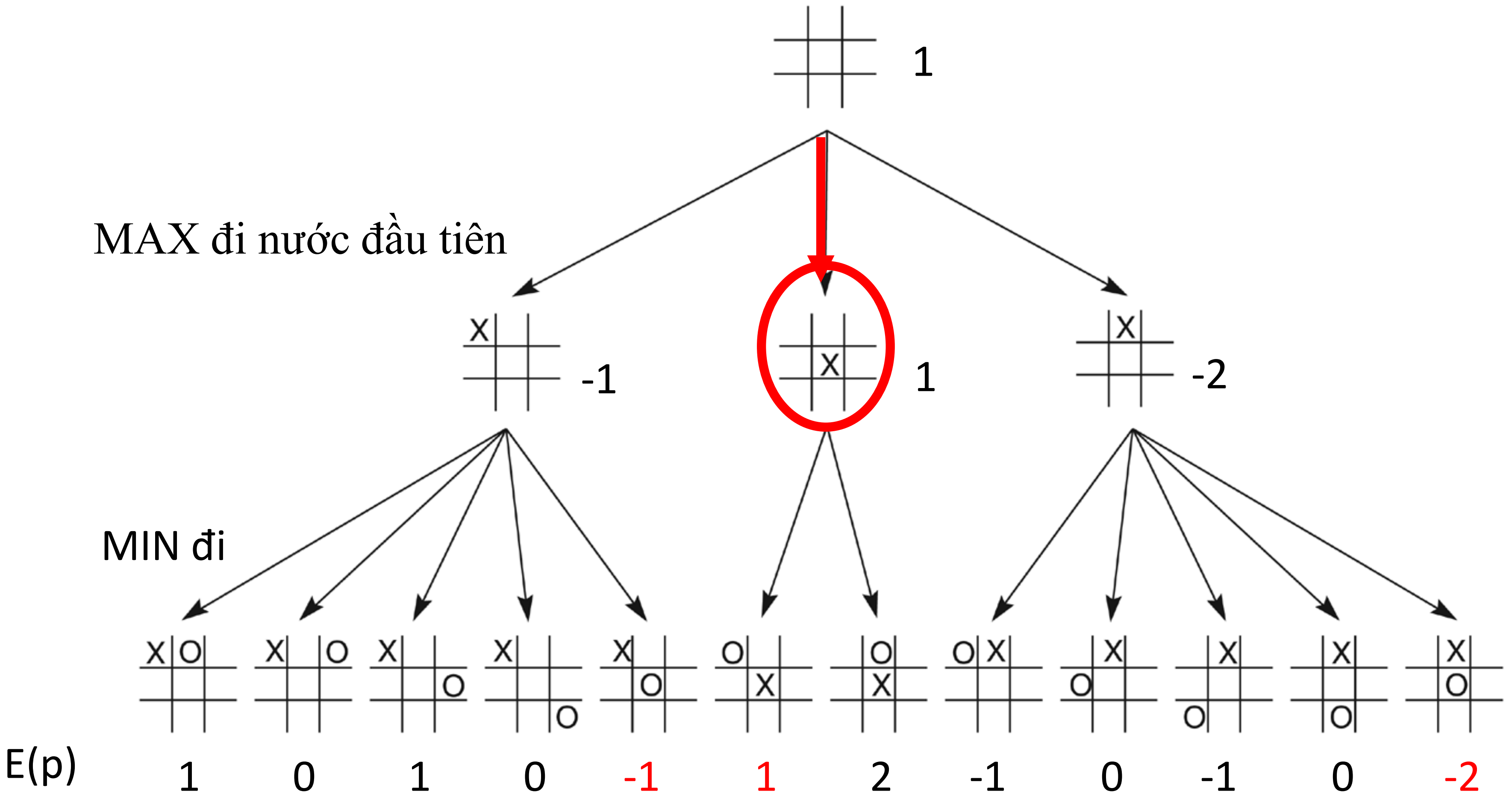
- Nếu một đỉnh ứng với nước đi của MAX thì giá trị của nó là Giá trị cực đại của các đỉnh con.
- Nếu một đỉnh ứng với nước đi của MIN thì giá trị của nó là Giá trị cực tiểu của các đỉnh con.

Áp dụng vào Tic-tac-toe, kích thước 3x3. MAX đặt dấu "x", MIN đặt dấu "o". Ở mỗi nước đi, mỗi đối thủ xem trước 2 nước.

Ước lượng E(p) đối với mỗi thế cờ p:

E(p) = (số dòng, cột, đường chéo còn mở với MAX)
– (số dòng, cột, đường chéo còn mở với MIN)

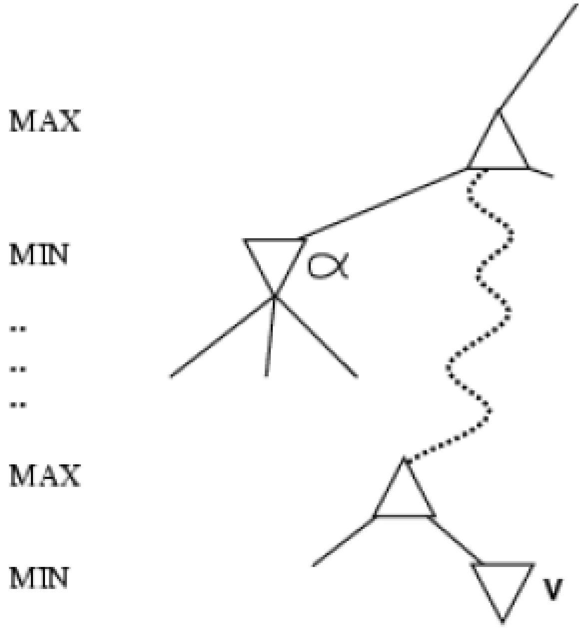
- Nếu p là thế thắng đối với MAX, E(p) = +∞
- Nếu p là thế thắng đối với MIN, E(p) = –∞
- MAX đi mọi đường không có "o"; MIN đi mọi đường không có "x"



→ Tìm kiếm theo kiểu depth-first

Phương pháp cắt tỉa α-β trong trò chơi minimax

- α là giá trị của lựa chọn tốt nhất được tìm thấy ở thời điểm hiện tại trên đường đi của max
- Nếu v tồi hơn α, max sẽ không duyệt nó → cắt tỉa nhánh đó
- Định nghĩa β tương tự đối với min

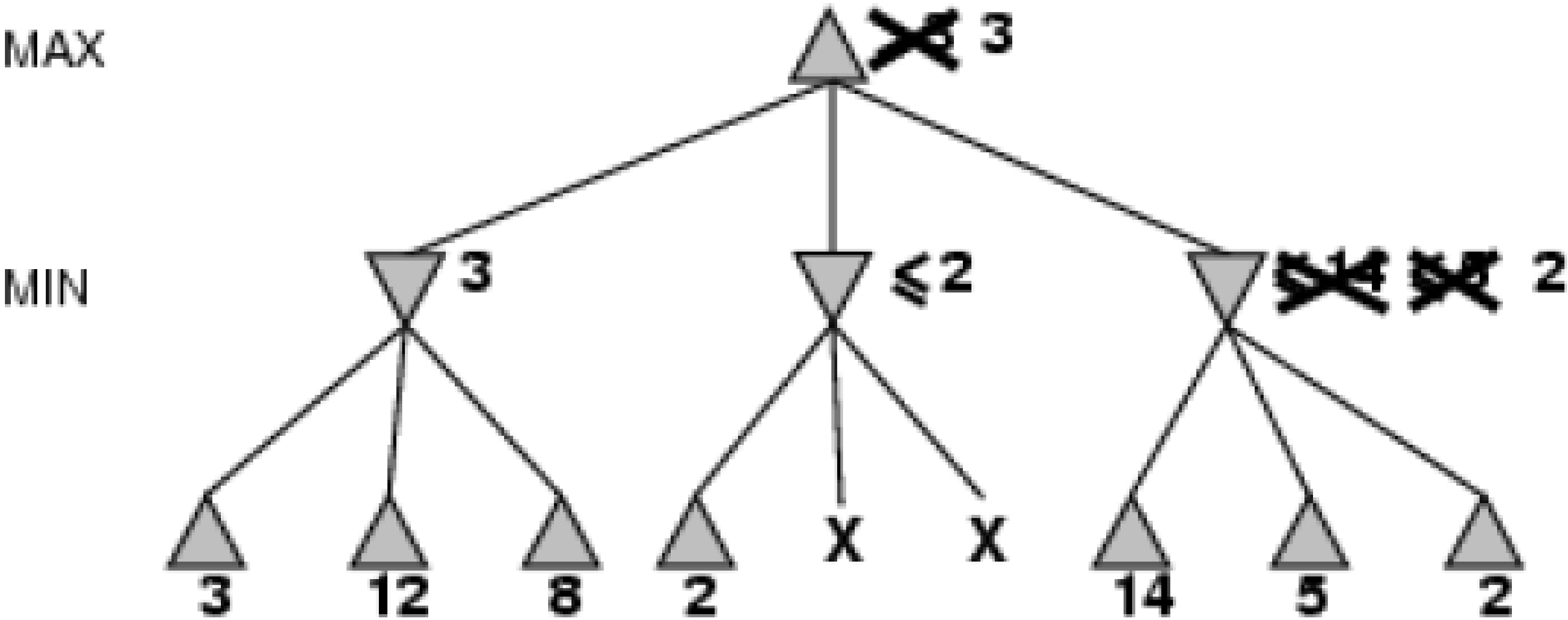


Cắt tỉa không làm ảnh hưởng tới kết quả cuối cùng

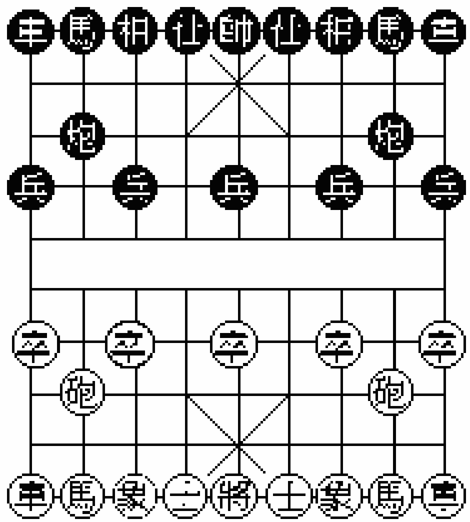
Sắp xếp thứ tự duyệt tối ưu sẽ nâng cao hiệu quả của quá trình cắt tỉa

Trong trường hợp tốt nhất, độ phức tạp thời gian = $O(b^m/2)$

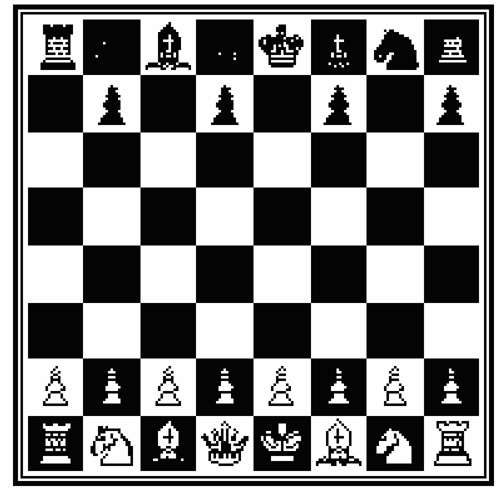
Phương pháp cắt tỉa α-β



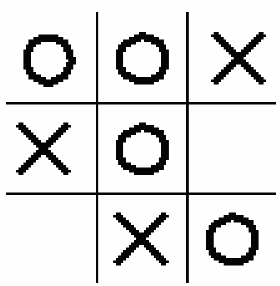
Một số trò chơi đối kháng (minimax)



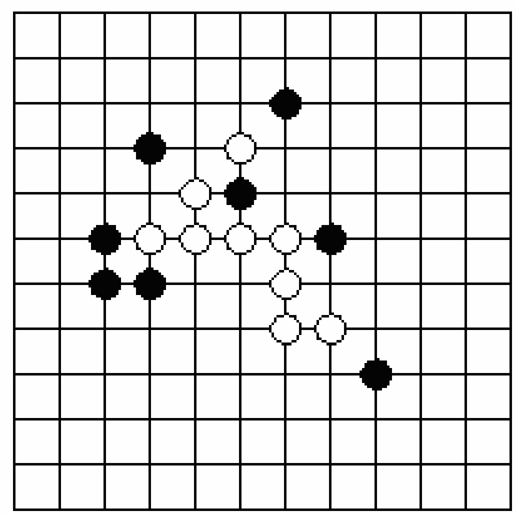
a) Cờ Tướng



b) Cờ Vua (cờ Quốc Tế)



c) Tictactoe



d) Go-moku (cờ caro)

3.4. CÁC PHƯƠNG PHÁP KHÁC

- Phương pháp thử và sai
- Phương pháp giải quyết bài toán tổng quát
- Phương pháp dựa trên thỏa mãn ràng buộc

Phương pháp thử và sai (test & set)

- Xuất phát từ n_0 : Mở = $\{n_0\}$; Đóng = $\varnothing$
- Tại mỗi thời điểm, chọn $n \in$ Mở để xét:

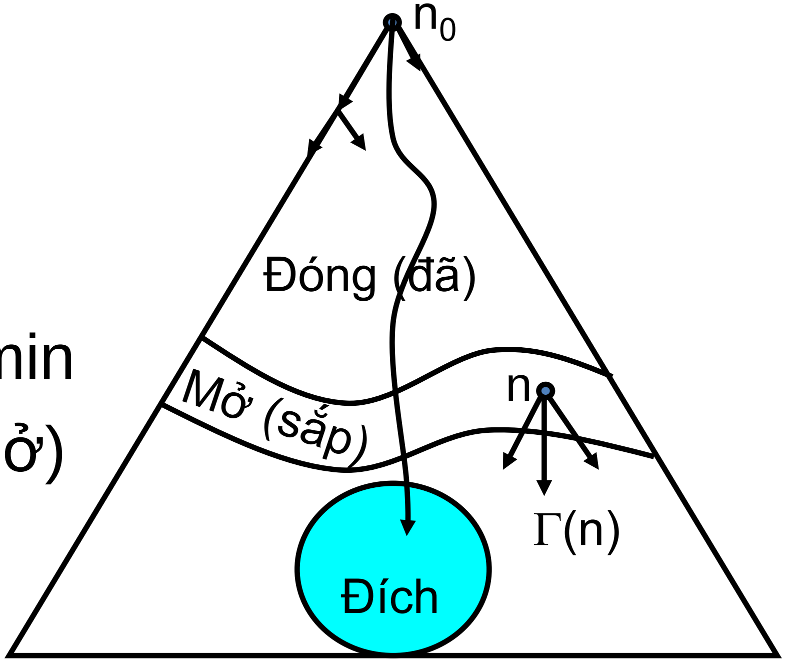
  Cách chọn !
- Mở = queue: $d(n) \rightarrow \min$

  Mở = stack: $d(n)$?

  TKCT: $g(n) = c(n0,n) \rightarrow \min$

  TKCT *: $f(n) = g(n) + h(n) \rightarrow \min$

  Thử và sai: $n \leftarrow$ get random(Mở)



Phương pháp giải quyết bài toán tổng quát (GPS)

GPS: Cách chọn $n \in$ Mở

- Với $\forall n \in$ Mở, xác định **sự khác biệt giữa n và Đích**:

  $\Delta = \{\delta_1, \ldots, \delta_m\}$
- Chọn sự khác biệt quan trọng nhất δ_i.
- Chọn biện pháp O_j phù hợp để giảm sự khác biệt δ_j bằng cách:
  - Xác định tập các phép biến đổi (toán tử) trong không gian

    $O=\{O_1, \ldots, O_n\}$
  - Xây dựng ma trân M với các cột là các toán tử, các hàng là các sự khác biệt: $M = (m_{ij})$, $i=1\div m$, $j = 1\div n$

$$m_{ij} = \begin{cases} 1 \text{ nếu } O_j \text{ làm giảm } \delta_i \\ 0 \text{ nếu ngược lại} \end{cases}$$

Nhận xét: giảm sự khác biệt với đích !

TKCT*: $n \in$ Mở : $f(n) = g(n) + h(n) \rightarrow \min$!

BÀI TOÁN THỎA MÃN RÀNG BUỘC

- Bài toán: Cho X = {X_1, …, X_n} là tập các biến, với các miền trị {D_1, …, D_n} tương ứng , C là tập các ràng buộc, mỗi ràng buộc là bộ đôi <*các biến tham gia*, *mối quan hệ*>, cần gán giá trị cho các biến {X_i=v_i, X_j=v_j, ...}, sao cho thỏa mãn các ràng buộc.
- Biểu diễn bài toán:

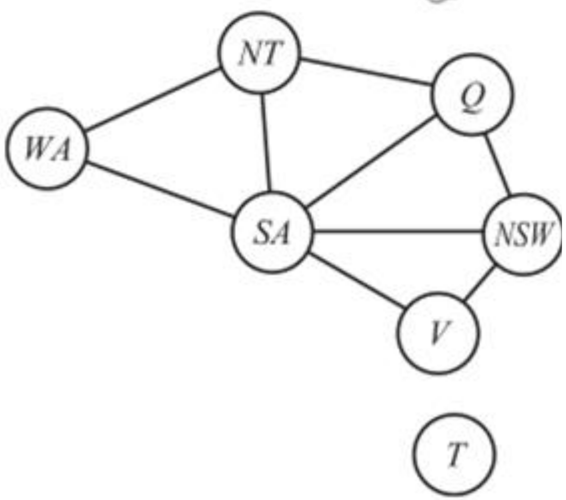
  - Không gian trạng thái: thể hiện các gán trị cho các biến

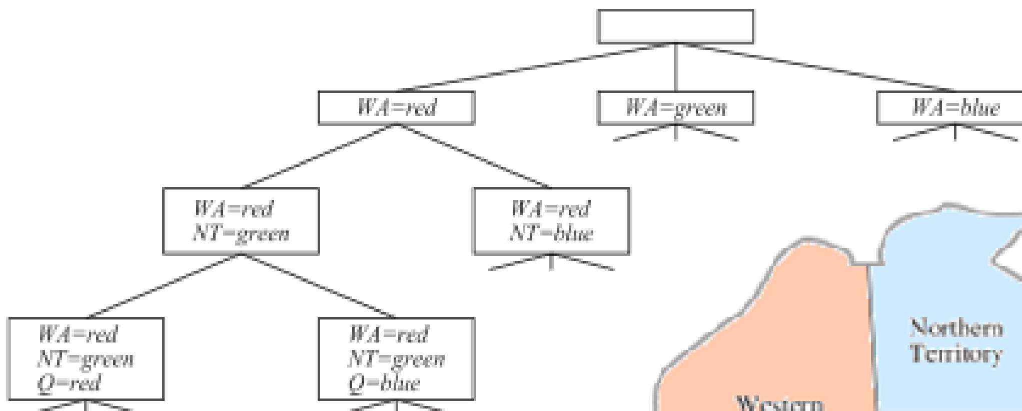
  - Đồ thị ràng buộc: thể hiện mối quan hệ cần thỏa mãn khi gán trị cho các biến

  - Lời giải của bài toán: là tập các cặp <*biến*, *giá trị*> thỏa mãn các ràng buộc

VÍ DỤ – BÀI TOÁN TÔ MÀU

- Cho X = {WA, NT, Q, NSW, V, SA, T} tô các màu D = {red, green, blue} sao cho cạnh nhau có màu khác nhau
- C = {SA≠WA, SA≠NT, SA≠Q, SA≠NSW, SA≠V, WA≠NT, NT≠Q, Q≠NSW, NSW≠V} → Đồ thị các ràng buộc





Phương pháp thỏa mãn ràng buộc

Mục đích: Tìm các trạng thái bài toán sao cho thỏa mãn tập ràng buộc nào đó

Quá trình tìm lời giải gồm hai phần:

- Tìm kiếm trong KGBT các ràng buộc
- Tìm kiếm trong KGBT các đỉnh

Nội dung:

Thực hiện các Bước 1 $\rightarrow$ 5 cho đến khi tìm được lời giải đầy đủ hoặc khi tất cả các đường đều đã duyệt nhưng không cho kết quả.

1. Cho một đỉnh n $\in$ MO
2. Áp dụng các luật suy dẫn với các ràng buộc vào đỉnh đã chọn để sinh ra tất cả các ràng buộc mới có thể có
3. Nếu tập các ràng buộc mới có mâu thuẫn $\rightarrow$ thông báo đường đi hiện tại đi vào ngõ cụt
4. Nếu tập ràng buộc mô tả lời giải đầy đủ của bài toán $\rightarrow$ dừng, thông báo "thành công". Ngược lại thực hiện tiếp bước 5.
5. Áp dụng các luật trong KGTT, tạo các lời giải bộ phận phù hợp với tập các ràng buộc hiện thời. Thêm các lời giải bộ phận này vào đồ thị tìm kiếm.

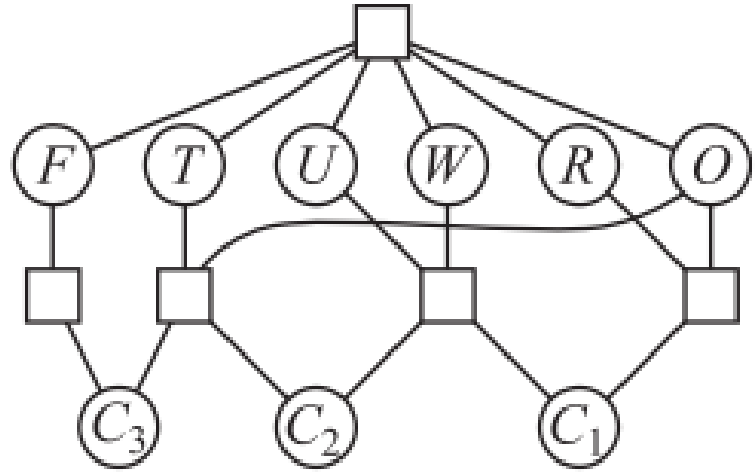
Thủ tục tìm kiếm Backtracking

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return BACKTRACK({ }, csp)

function BACKTRACK(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
     if value is consistent with assignment then
        add {var = value} to assignment
        inferences ← INFERENCE(csp, var, value)
        if inferences ≠ failure then
          add inferences to assignment
          result ← BACKTRACK(assignment, csp)
          if result ≠ failure then
            return result
     remove {var = value} and inferences from assignment
  return failure
```

VÍ DỤ

$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$



O + O = R + 10.C1

W + W + C1 = U + 10.C2

T + T + C2 = O + 10.C3

C3 = F

F, T, U, W, R, O $\in$ {0,1,2,3,4,5,6,7,8,9}, “khác nhau”

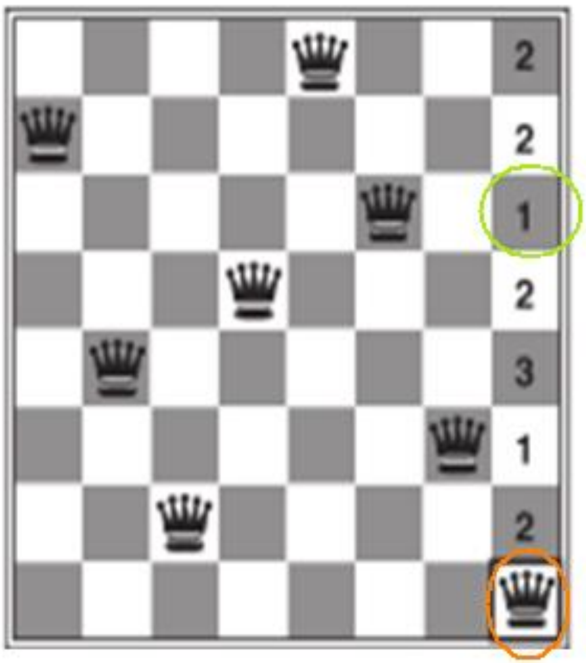
C1, C2. C3 $\in$ {1, 0}   TWO=846, FOUR=1692

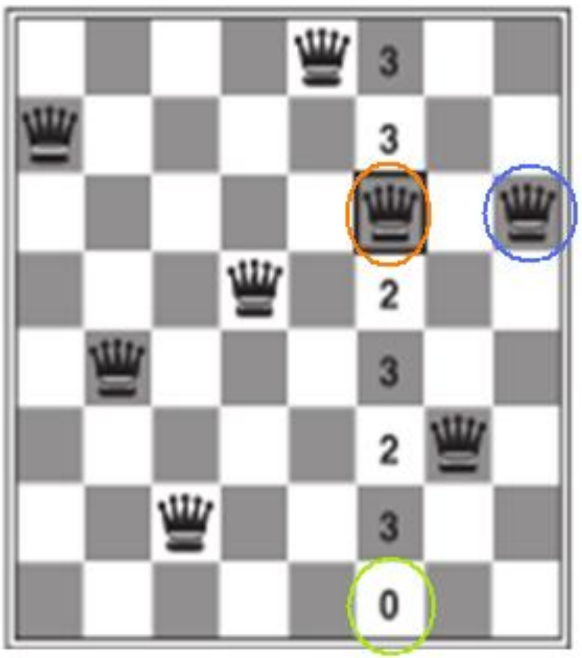
Thủ tục Local search

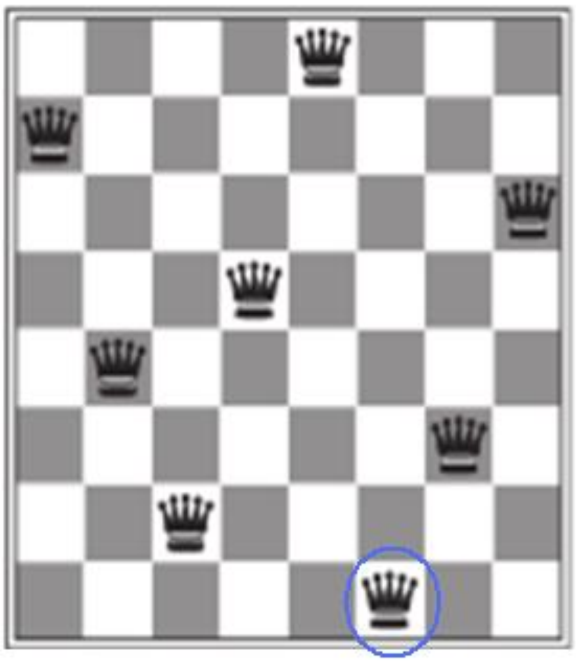
```
function MIN-CONFLICTS(csp, max_steps) returns a solution or failure
   inputs: csp, a constraint satisfaction problem
           max_steps, the number of steps allowed before giving up

   current ← an initial complete assignment for csp
   for i = 1 to max_steps do
      if current is a solution for csp then return current
      var ← a randomly chosen conflicted variable from csp.VARIABLES
      value ← the value v for var that minimizes CONFLICTS(var, v, current, csp)
      set var = value in current
   return failure
```

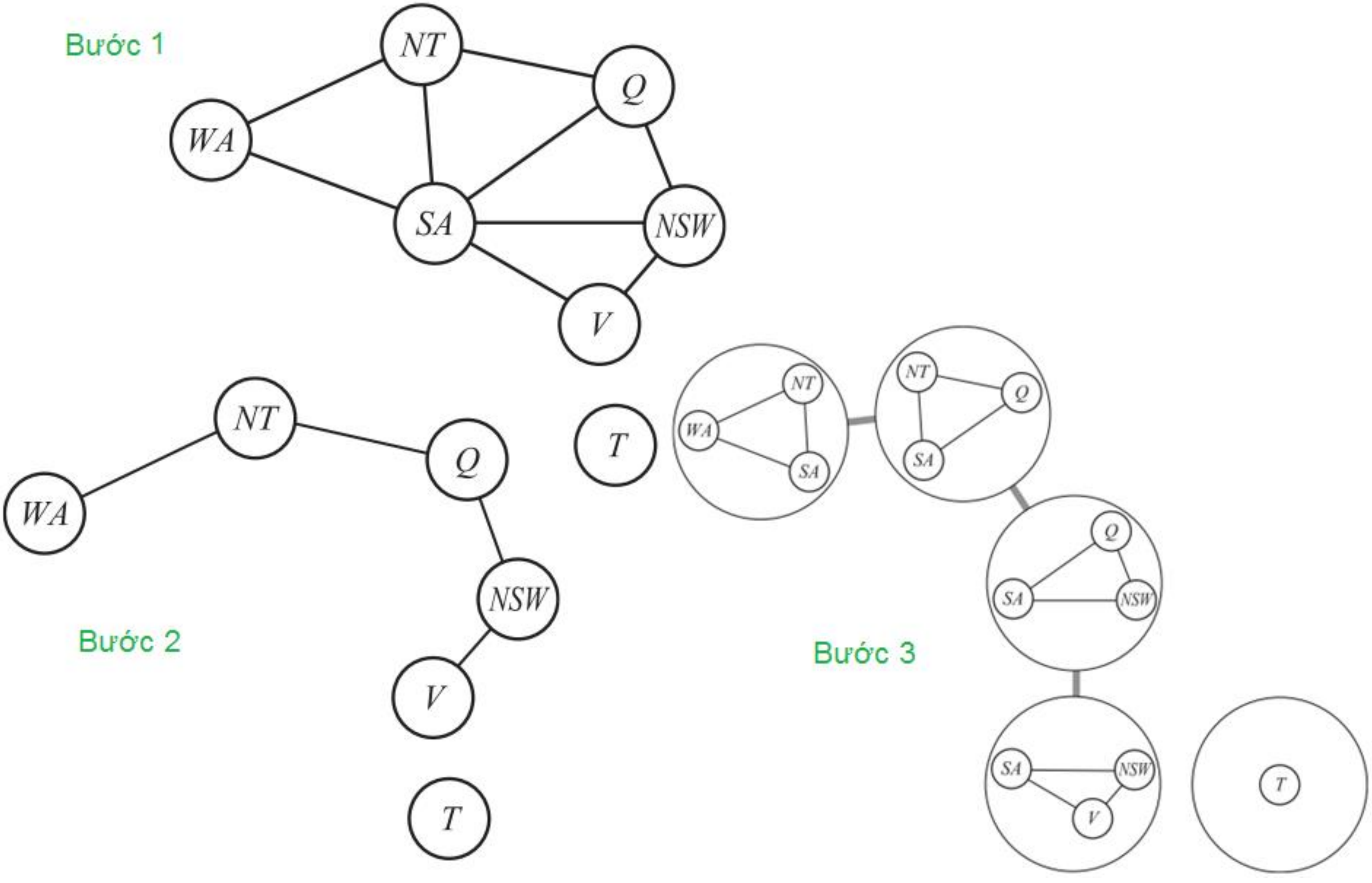
Ví dụ
Bài toán 8 quân hậu







Chuyển đổi cấu trúc bài toán



Giải thuật di truyền

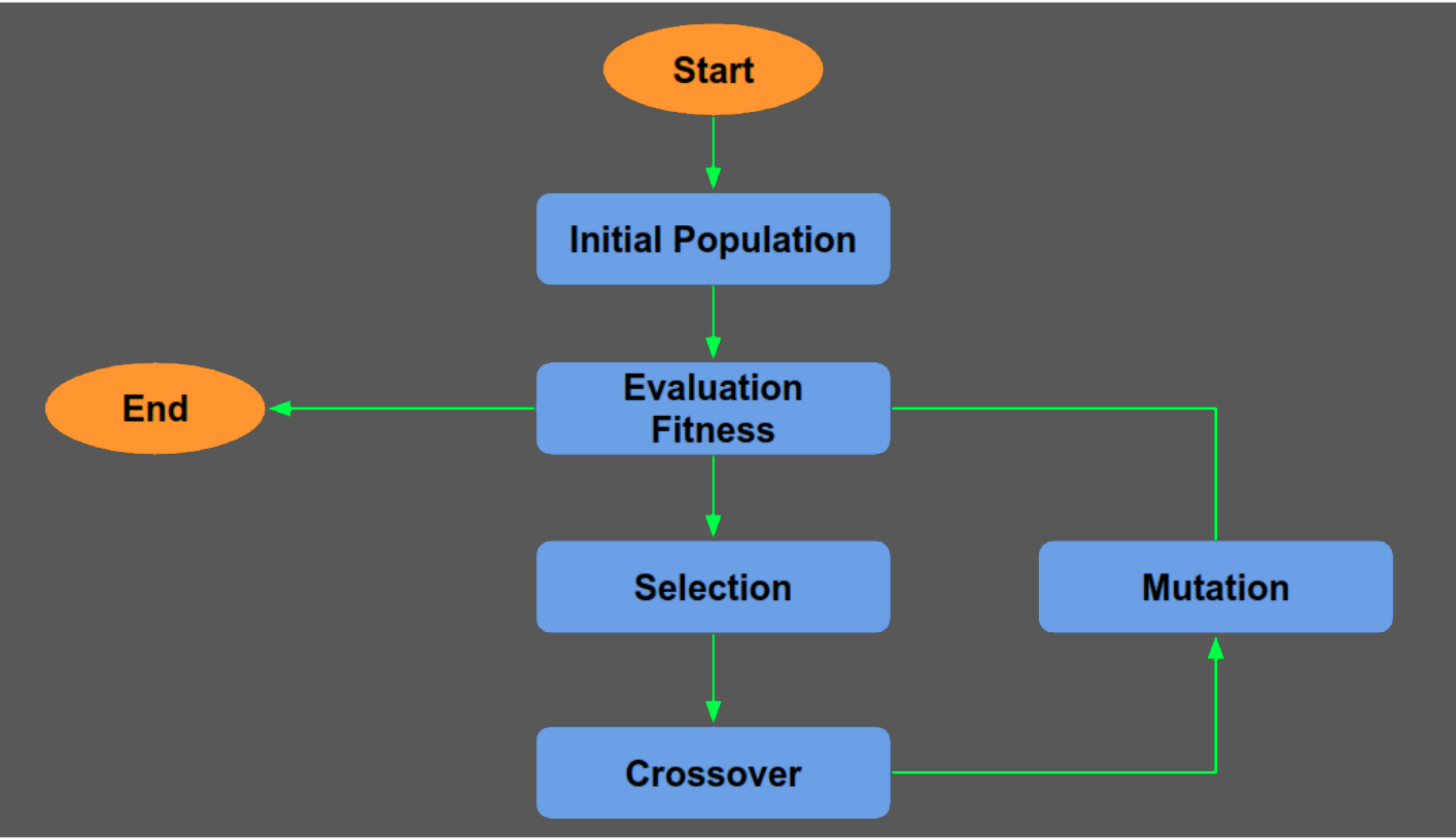
- MỞ ~ Quần thể
- Đánh giá n $\in$ MỞ ~ Độ thích nghi của n với Đích muốn đạt được
- Ở mỗi bước lặp, thực hiện

Lai ghép: từ hai cá thể, lai ghép thành các cá thể mới

Đột biến: thay đổi vài gen của cá thể

Chọn lọc: giữ lại các cá thể có độ thích nghi tốt

- Cho đến khi xuất hiện cá thể thích nghi “tốt” với Đích


Start
Initial Population
Evaluation
Fitness
End
Selection
Mutation
Crossover