

# **PROCURA EM ESPAÇOS DE ESTADOS**

(PARTE 3)

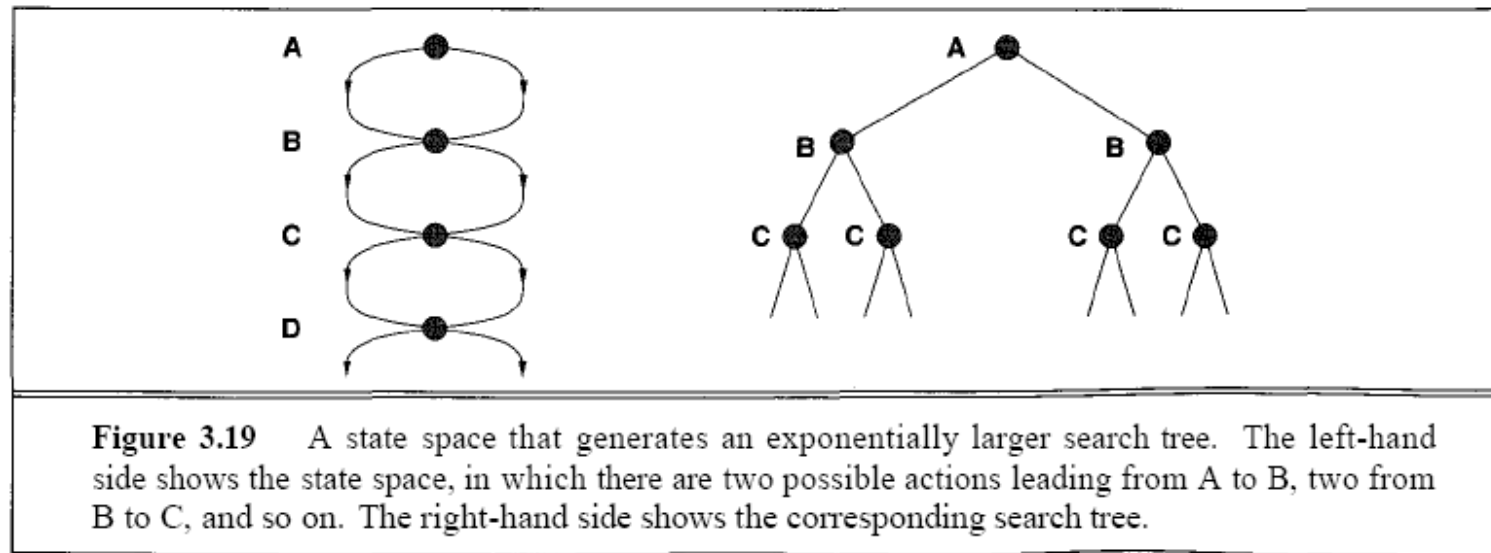
Luís Morgado

2015

# PROCURA EM GRAFOS COM CICLOS

## ESTADOS REPETIDOS NA ÁRVORE DE PROCURA

- Acontece quando as **acções** correspondentes às transições de estado são **reversíveis**
- o grafo do espaço de estados apresenta **ciclos**



## EXPANSÃO DE ESTADOS JÁ ANTERIORMENTE ANALISADOS

- **Desperdício de recursos (tempo, memória)**

# PROCURA EM GRAFOS COM CICLOS

## MEMÓRIA DE NÓS PROCESSADOS

- Nós gerados mas **não expandidos**  
(**fronteira** de exploração)
  - **ABERTOS**
- Nós **expandidos**
  - **FECHADOS**

# PROCURA EM GRAFOS COM CICLOS

```
function GRAPH-SEARCH(problem, fringe) returns a solution, or failure  
  
  closed ← an empty set  
  fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)  
  loop do  
    if EMPTY?( fringe) then return failure  
    node ← REMOVE-FIRST(fringe)  
    if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)  
    if STATE[node] is not in closed then  
      add STATE[node] to closed  
      fringe ← INSERT-ALL(EXPAND(node, problem), fringe)
```

**Figure 3.19** The general graph-search algorithm. The set *closed* can be implemented with a hash table to allow efficient checking for repeated states. This algorithm assumes that the first path to a state *s* is the cheapest (see text).

[Russel & Norvig, 2003]

# PROCURA GERAL EM GRAFOS

- Ao gerar novo nó sucessor *noSuc* é necessário considerar:
  - $noSuc \notin Abertos \wedge noSuc \notin Fechados$ 
    - Inserir *noSuc* em *Abertos*
  - $noSuc \in Abertos$ 
    - Se *noSuc* foi atingido através de um caminho mais curto (com menor custo)
      - Remover nó anterior de *Abertos*
      - inserir *noSuc* em *Abertos*
  - $noSuc \in Fechados$ 
    - Se *noSuc* foi atingido através de um caminho mais curto (com menor custo)
      - Remover nó anterior de *Fechados*
      - inserir *noSuc* em *Abertos*

# PROCURA EM GRAFOS COM CICLOS

## MEMÓRIA DE NÓS PROCESSADOS

- Nós gerados mas **não expandidos**  
(**fronteira** de exploração)

- **ABERTOS**

- Nós **expandidos**

- **FECHADOS**

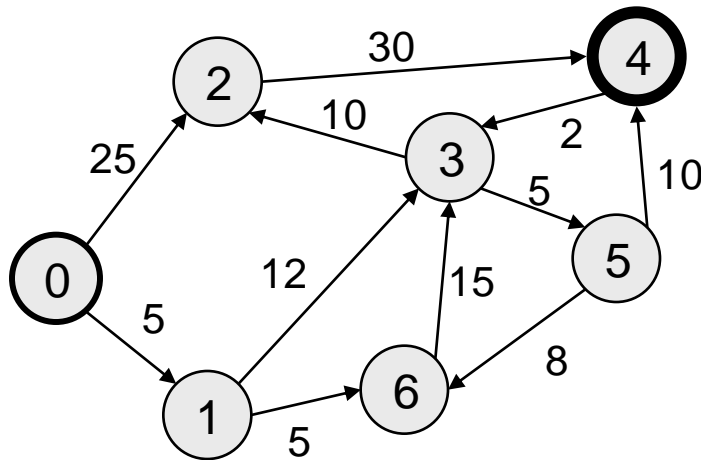
**EXPLORADOS**

# PROCURA MELHOR-PRIMEIRO (*BEST-FIRST*)

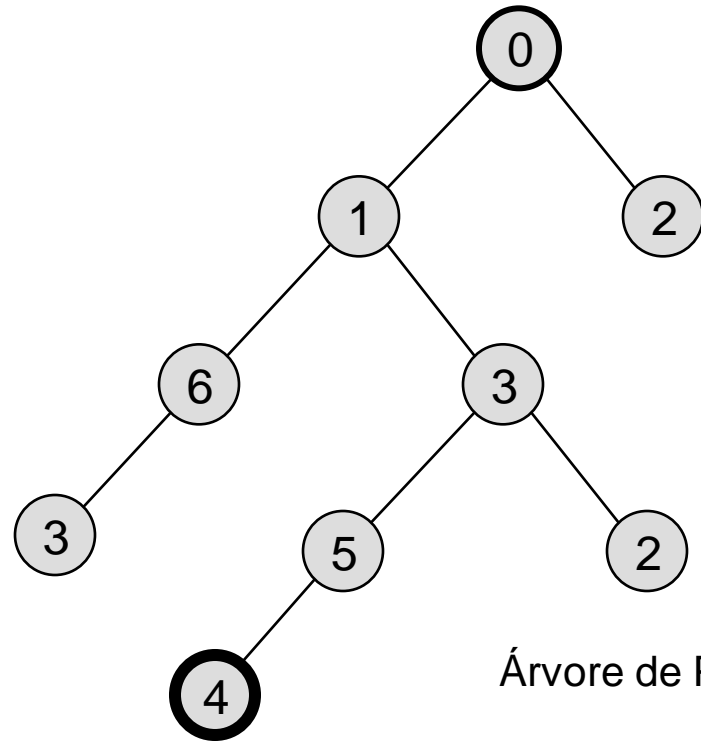
- Tira partido de uma **avaliação do estado**
- Utiliza uma função  **$f$**  para **avaliação** de cada nó  $n$  gerado
  - $f(n) \geq 0$
  - Tipicamente  $f(n)$  representa uma estimativa do custo da solução através do nó  $n$ 
    - Quanto menor o valor de  $f(n)$  mais promissor é o nó  $n$
- A **fronteira de exploração** (*Fringe / Abertos*) é **ordenada por ordem crescente de  $f(n)$**

# PROCURA DE CUSTO UNIFORME

- Estratégia de controlo
  - Explorar primeiro caminhos com menor custo
  - Custo de transição  $\geq \varepsilon > 0$



Grafo do  
Espaço de Estados



Árvore de Procura



# **MÉTODOS DE PROCURA NÃO INFORMADA**

## **PROCURA EM PROFUNDIDADE**

- Critério de exploração: maior profundidade
- Variantes
  - **PROCURA EM PROFUNDIDADE LIMITADA**
  - **PROCURA EM PROFUNDIDADE ITERATIVA**

## **PROCURA EM LARGURA**

- Critério de exploração: menor profundidade
- Variantes
  - **PROCURA BIDIRECCIONAL**

## **PROCURA DE CUSTO UNIFORME**

- Critério de exploração: custo de transição  $> 0$

# COMPLEXIDADE COMPUTACIONAL

Método de Procura	Tempo	Espaço	Óptimo	Completo
Profundidade	$O(b^m)$	$O(bm)$	Não	Não
Largura	$O(b^d)$	$O(b^d)$	Sim	Sim
Custo Uniforme	$O(b^{\lceil C^*/\varepsilon \rceil})$	$O(b^{\lceil C^*/\varepsilon \rceil})$	Sim	Sim
Profundidade Limitada	$O(b^l)$	$O(bl)$	Não	Não
Profundidade Iterativa	$O(b^d)$	$O(bd)$	Sim	Sim
Bidireccional	$O(b^{d/2})$	$O(b^{d/2})$	Sim	Sim

$b$  – factor de ramificação

$d$  – dimensão da solução

$m$  – profundidade da árvore de procura

$l$  – limite de profundidade

$C^*$  – Custo da solução óptima

$\varepsilon$  – Custo mínimo de uma transição de estado ( $\varepsilon > 0$ )

# BIBLIOGRAFIA

[Russel & Norvig, 2009]

S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, 2009

[Nilsson, 1998]

N. Nilsson , *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann 1998

[Luger, 2009]

G. Luger , *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* , Addison-Wesley, 2009

[Jaeger & Hamprecht, 2010]

M. Jaeger, F. Hamprecht, *Automatic Process Control for Laser Welding*, Heidelberg Collaboratory for Image Processing (HCI) , 2000