

# **PROCURA EM ESPAÇOS DE ESTADOS**

Luís Morgado

2021

# Procura em Espaços de Estados

---

- **Métodos de Procura não Informada**

- Estratégias de exploração do espaço de estados (*controlo da procura*) **não tiram partido de *conhecimento do domínio do problema*** para ordenar a fronteira de exploração
- **Procura não guiada** (exaustiva)

- **Métodos de Procura Informada**

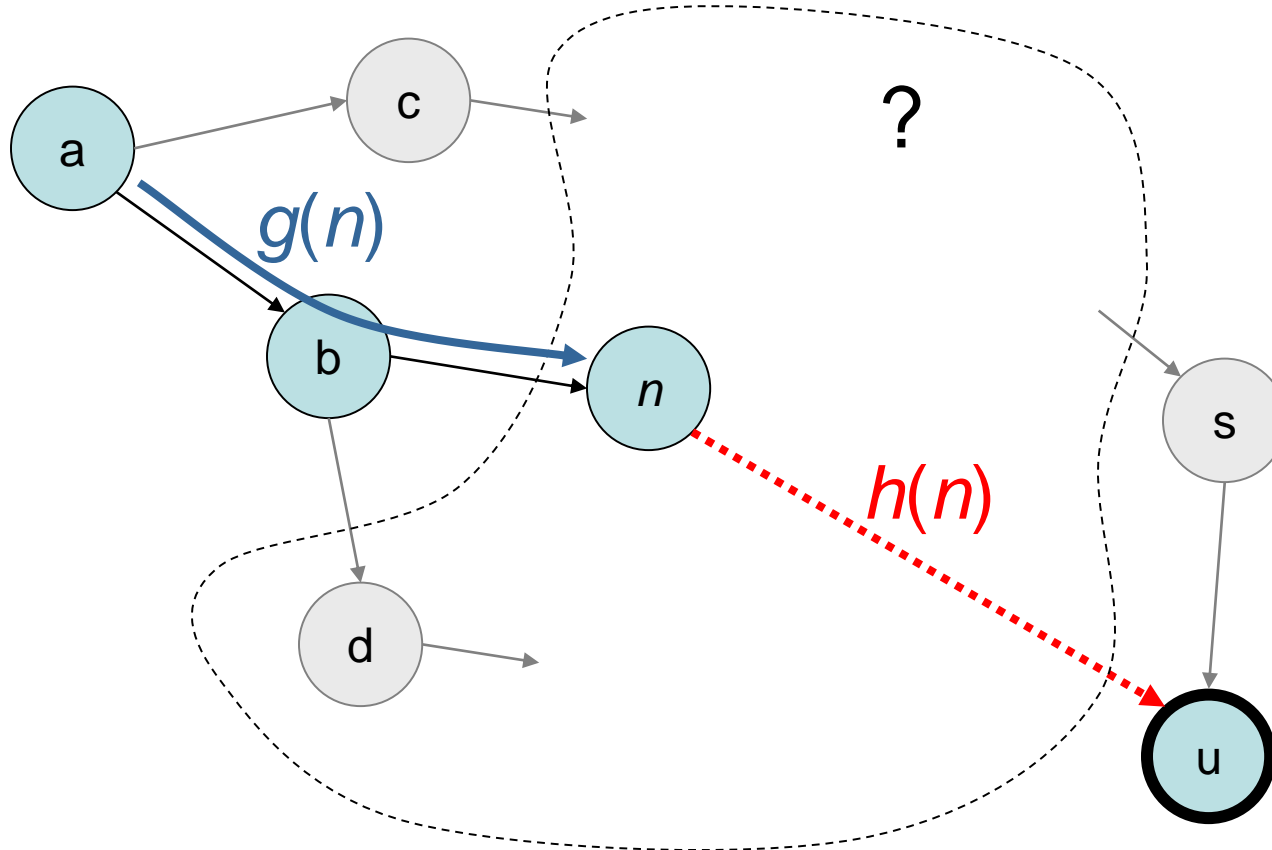
- Estratégias de exploração do espaço de estados (*controlo da procura*) **tiram partido de *conhecimento do domínio do problema*** para ordenar a fronteira de exploração
- **Procura guiada** (selectiva)

# Função heurística $h(n)$

---

- Representa uma **estimativa do custo do percurso desde o nó  $n$  até ao nó objectivo**
- Reflecte conhecimento acerca do domínio do problema, para guiar a procura
- O seu valor é independente do percurso até  $n$ 
  - Depende apenas de:
    - **Estado** associado a  $n$
    - **Objectivo**

# Métodos de Procura Informada



$g(n)$  – Custo do percurso até  $n$

$h(n)$  – Estimativa de custo de  $n$  até ao objectivo

# Procura Melhor-Primeiro (*Best-First*)

---

- Utiliza uma função  **$f(n)$**  para **avaliação** de cada nó  **$n$**  gerado
  - **$f(n) \geq 0$**
  - **$f(n)$**  representa uma **avaliação** 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** é **ordenada por ordem crescente de  $f(n)$**

# Procura Melhor-Primeiro (*Best-First*)

---

- 3 variantes principais
  - $f(n) = g(n)$ 
    - **Procura de Custo Uniforme**
      - Não tira partido de conhecimento do domínio do problema expresso através da função  $h(n)$
  - $f(n) = h(n)$ 
    - **Procura Sôfrega (*Greedy Search*)**
      - Não tem em conta o custo do percurso explorado
      - Minimização de custo **local**
      - Soluções **sub-óptimas** (problema dos óptimos locais)
  - $f(n) = g(n) + h(n)$ 
    - **Procura A\*** (heurística admissível)
      - Minimização de custo **global**

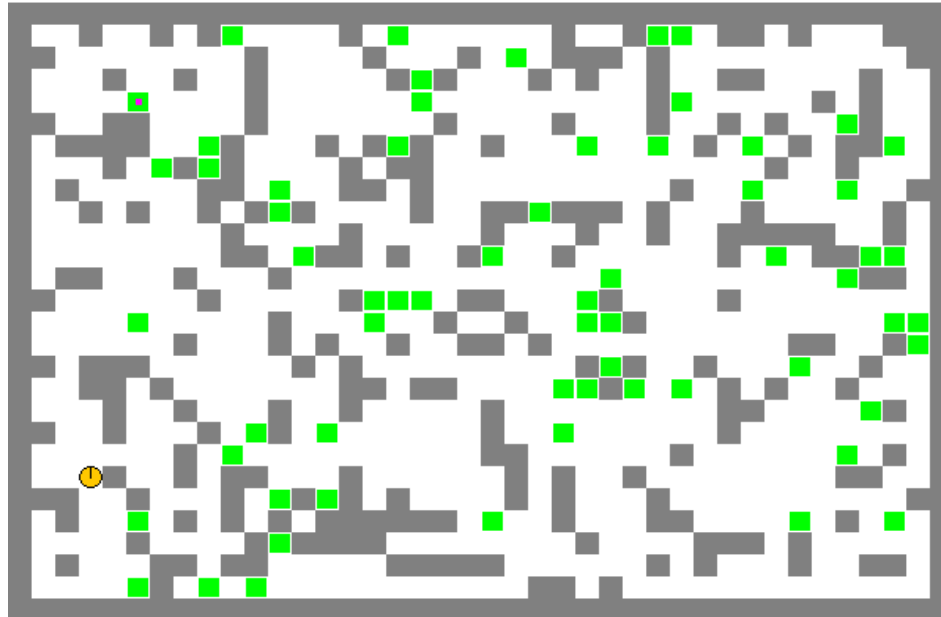
# Procura $A^*$

---

- **Heurística admissível**
  - $0 \leq h(n) \leq h^*(n)$
  - $h^*(n)$ 
    - Custo mínimo do nó  $n$  até ao objectivo (percurso óptimo)
- Uma heurística **admissível** é **optimista**
  - A estimativa de custo é **sempre inferior ou igual** ao custo efectivo **mínimo**
  - Para um nó objectivo  $n_{\text{obj}}$ 
    - $h(n_{\text{obj}}) = 0$

# Procura A\*

---



$h_1$  – Distância Euclidiana

$$h_1(n) = \sqrt{(x_n - x_{obj})^2 + (y_n - y_{obj})^2}$$

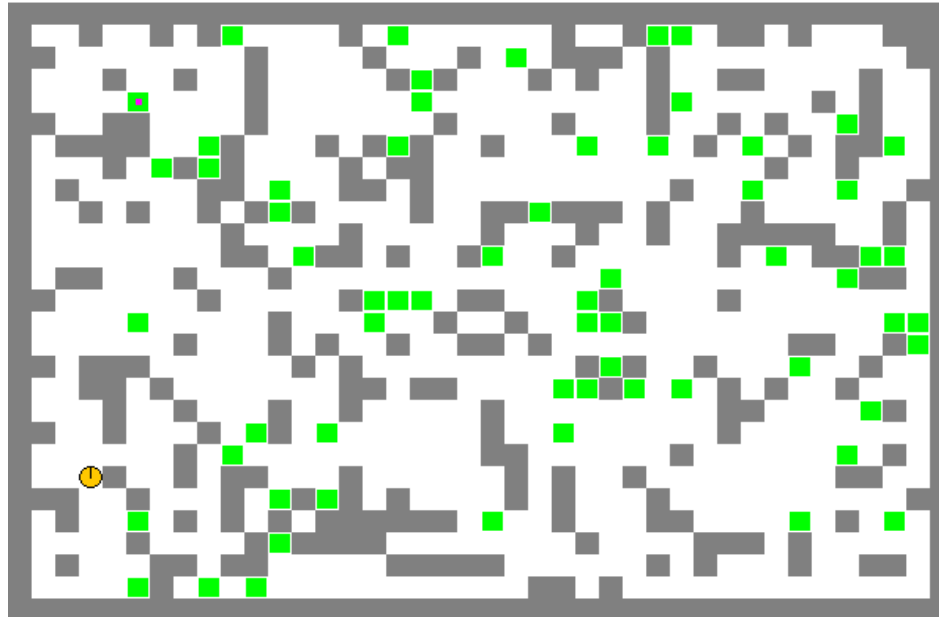
Admissível?

**SIM**



# Procura A\*

---



$h_2$  – Distância de Manhattan

$$h_2(n) = |x_n - x_{obj}| + |y_n - y_{obj}|$$

**Admissível?**

- SIM : Se não forem possíveis movimentos diagonais
- NÃO : Caso contrário

# Procura A\*

---

- Como definir uma heurística admissível
  - No caso geral, uma **heurística admissível** é obtida através da **remoção de restrições** associadas ao problema
  - Exemplo: Navegação autónoma
    - $h_1$  - Distância de Manhattan
      - Corresponde a retirar a restrição:
        - » Não movimentação através de obstáculos
    - $h_2$  - Distância de Euclidiana
      - Corresponde a retirar as restrições:
        - » Não movimentação através de obstáculos
        - » Não movimentação em diagonal

# Procura A\*

---

- **$C^*$**  - Custo da solução óptima
- **$n$**  - Nó na fronteira de exploração  
$$f(n) = g(n) + h(n) \leq C^* \text{ (se } h(n) \text{ admissível)}$$
- **$m$**  - Nó sub-óptimo na fronteira de exploração  
$$f(m) = g(m) + h(m)$$
- Se  **$m$**  for um nó objectivo  
$$h(m) = 0$$
$$f(m) = g(m) > C^*$$
- Então  
$$f(n) \leq C^* < f(m)$$

**$m$  não será expandido e a solução encontrada será óptima**

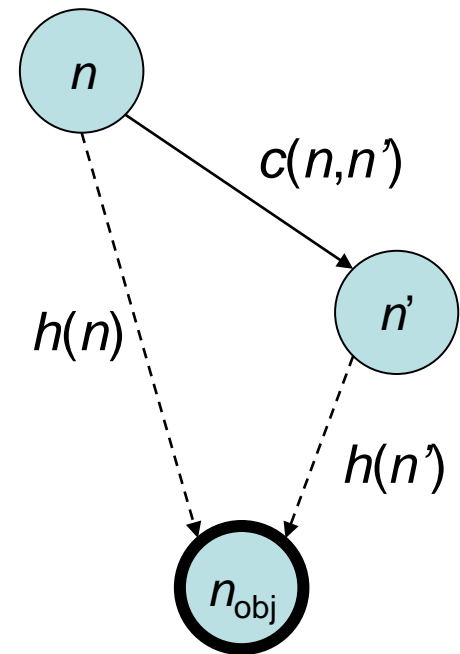
# Procura A\*

---

- O método de procura A\* é
  - Completo
  - Óptimo
- **Se os nós já visitados não forem eliminados**
  - **A heurística pode não ser consistente**

# Procura A\*

- **Heurística consistente** (ou monótona)
  - Para cada nó  $n$ , seu sucessor  $n'$  e custo de transição  $c(n, n')$ 
    - $h(n) \leq c(n, n') + h(n')$
  - Para um nó objectivo
    - $h(n_{\text{obj}}) = 0$
- Uma heurística consistente é também admissível
- **Uma heurística admissível pode não ser consistente**



# Procura A\*

---

- Se  $h(n)$  for consistente os valores de  $f(n)$  nunca diminuem ao longo de um caminho

- Consideremos  $n'$  um sucessor de  $n$

$$g(n) + c(n, n') + h(n') \geq g(n) + h(n)$$



- Qualquer **nó seleccionado para expansão tem de estar num percurso óptimo**, pois qualquer outro caminho terá um custo no mínimo igual

# Procura A\*

---

- Com uma heurística consistente o método de procura A\* é
  - Completo
  - Óptimo
- **Mesmo se os nós já visitados forem eliminados**
  - Redução da complexidade da procura

# Procura A\* com heurística consistente

---

Ao gerar novo nó sucessor *noSuc*:

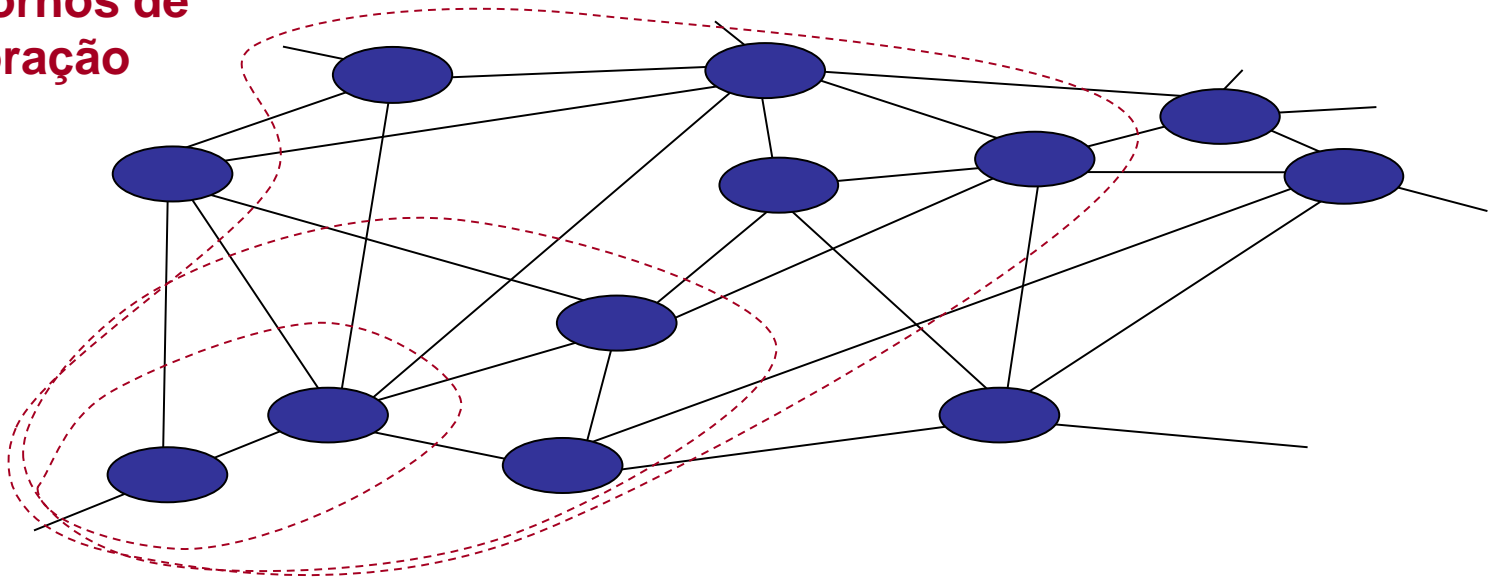
- *noSuc*  $\notin$  *Abertos*  $\wedge$  *noSuc*  $\notin$  *Fechados*
  - Inserir *noSuc* em *Abertos*
- *noSuc*  $\in$  *Abertos*
  - Se *noSuc* foi atingido através de um caminho mais curto
    - Remover nó anterior de *Abertos*
    - inserir *noSuc* em *Abertos*
- ***noSuc*  $\in$  *Fechados***
  - Eliminar *noSuc*



# Procura A\*

---

Contornos de  
exploração



- Para uma heurística consistente
  - Sempre que é expandido um nó o percurso desse nó é ótimo
  - São expandidos todos os nós com  $f(n) < C^*$
  - São eventualmente expandidos nós com  $f(n) = C^*$  antes do nó objectivo

# Procura A\*

---

- Método de procura de **eficiência óptima** para qualquer função heurística
  - Nenhum outro algoritmo expandirá menos nós, mantendo as características de ser **completo** e **óptimo**, excepto nas situações de escolha entre nós com  $f(n) = C^*$
- **No entanto, não resolve o problema da complexidade combinatória**
  - O número de nós expandidos dentro do contorno do nó objectivo continua a ser uma **função exponencial** da **dimensão do percurso** até ao objectivo
  - Função heurística afecta o contorno de procura
    - Pode não ser suficiente

# Referências

---

[Russel & Norvig, 2003]

S. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach”, 2nd Edition, Prentice Hall, 2003

[Pearl, 1984]

J. Pearl, “Heuristics: Intelligent Search Strategies for Computer Problem Solving”, Addison-Wesley, 1984