procura informada

procura informada – caraterização

- usa informação adicional

  para além da especificação do problema

- usa uma função de avaliação do estado: $f(n)$

  geralmente inclui uma componente **heurística**

  $h(n)$ = (menor) custo estimado de n até ao objetivo

  tem de verificar $h(\text{objetivo}) = 0$

  ex: navegador num mapa,

  h = distância em linha reta até objetivo

  ou

  h = nº de quarteirões (se numa cidade) ~ distância de Manhattan

procura sôfrega heurística pura (*greedy best-first*)

greedy best-first
e não
greedy

fonte: CS188 Berkeley

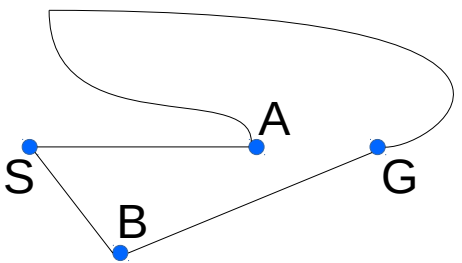
procura sôfrega heurística pura

- usa apenas a (heurística) estimativa de custo ao objetivo

  $f(n) = h(n)$

  expande o nó com o **menor custo estimado até ao objetivo**

ex: mapa de estradas, com heurística de distância mais curta em linha reta



claramente, usar apenas a heurística pode ter problemas
resultado: S-A-G
mas S-A-G > S-B-G

procura A*

fonte: CS188 Berkeley

procura A*

escolhe o nó com o menor valor de

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

i.e.

custo do caminho desde o início até ao nó

+

custo estimado do caminho até ao objetivo

$f(n)$ = custo estimado da melhor solução que passa por n

A* propriedades

- completo

  e

- garante ótimo

  desde que…

  b seja finito
  custo, *c*, das ações positivo
  $c > \delta > 0$

e a heurística, $h(n)$ seja

- admissível (EE árvore)

  e

- consistente (EE grafo)

admissibilidade

heurística admissível

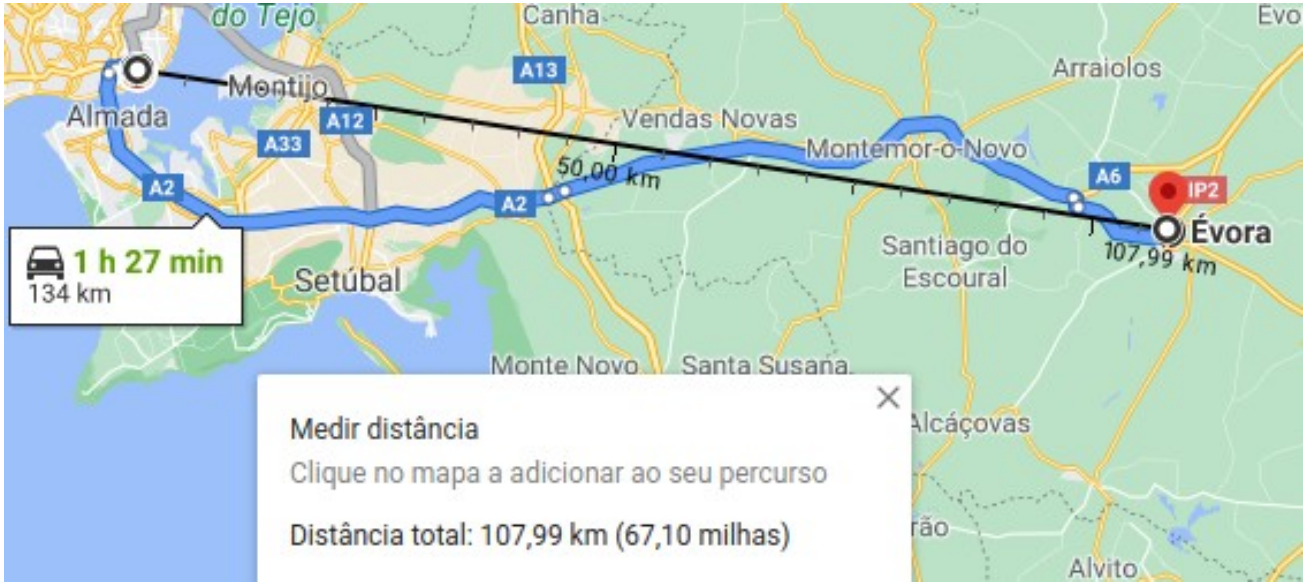
não negativa

nunca sobrestima o custo de atingir o objetivo

$0 \leq h(n) \leq h^*(n)$

$h^*(n)$: custo real até ao objetivo

ex: heurística de distância em linha reta, num mapa de estradas



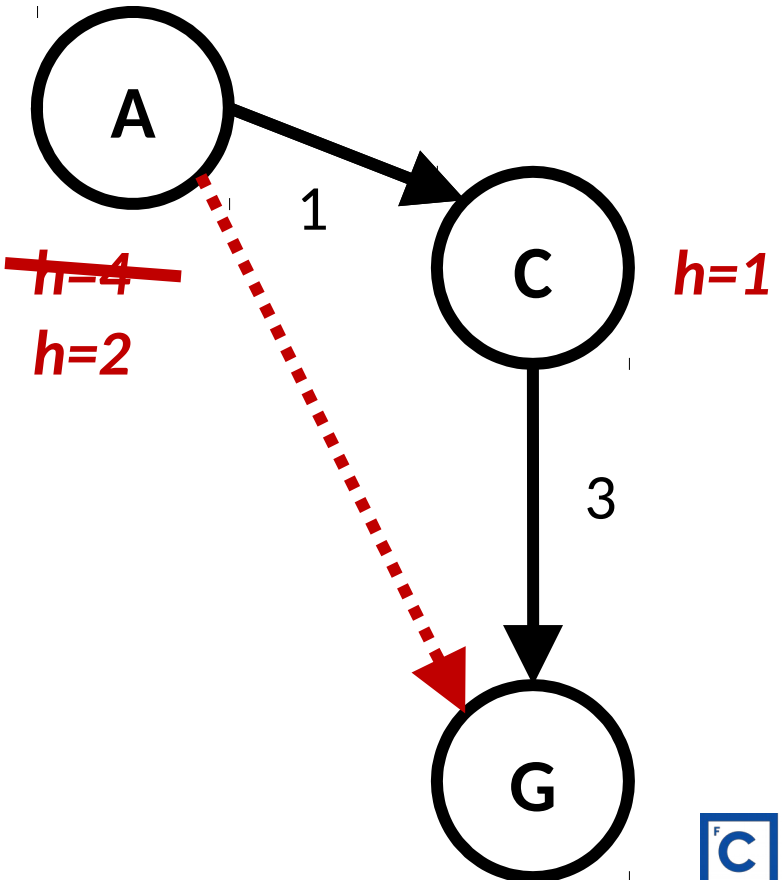
consistência

heurística consistente

$$h(n) \leq c(n, a, n') + h(n')$$

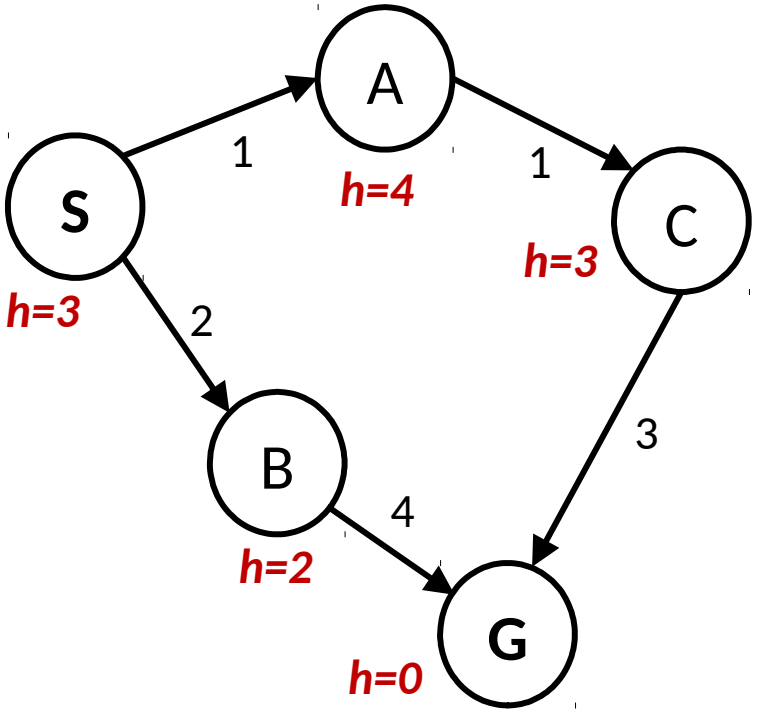
estimativa em *n* não pode exceder o custo da ação, *a*, que leva de *n* ao sucessor *n*', somado à estimativa em *n*'

(desigualdade triangular)

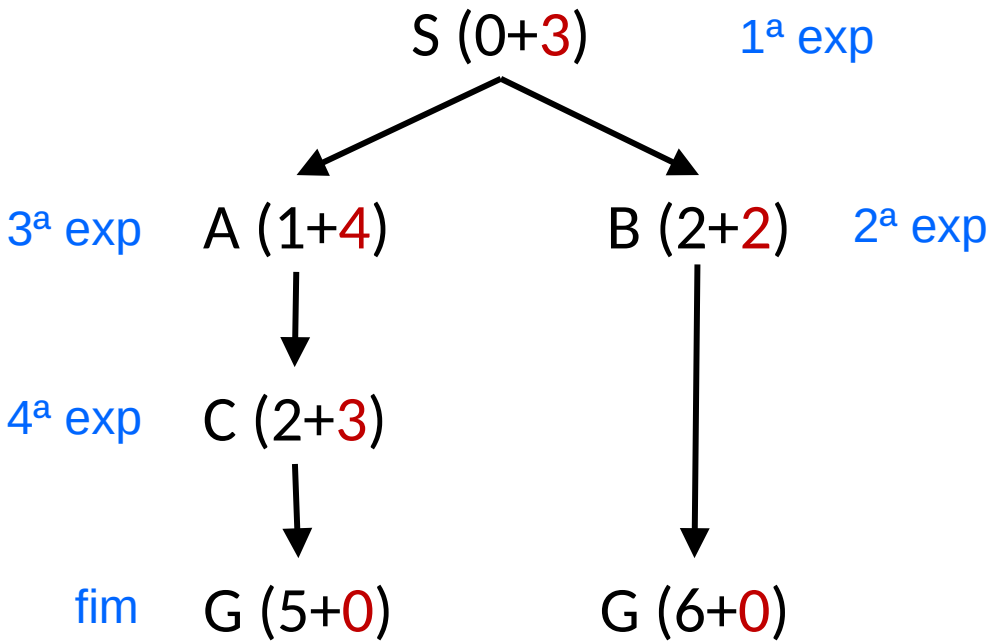


A* passo a passo

espaço de estados



árvore de procura



resultado: S – A – C – G

Ciências ULisboa

A* resultado

- comparar com PL, PP, PH
- verificar admissibilidade e consistência de $h(n)$
- observar a importância de uma boa heurística

a ver numa próxima sessão perto de si

A* paragem

- pára quando se vai expandir um nó objetivo

  (não quando se gera um nó objetivo)

  quando o nó a expandir é um objetivo garante-se que
  não há nenhum caminho mais económico a atingir um objetivo

A* expande todos os nós com $f(n) < C^*$ (C^* = custo ótimo)

A* não expande qualquer nó com $f(n) > C^*$

otimalidade

- procura em árvore:

  A* é optimal se a heurística for admissível

  custo uniforme é um caso especial (h = 0)

- procura em grafo:

  A* é optimal se a heurística for consistente

  custo uniforme é optimal (h = 0 é consistente)

- consistência implica admissibilidade

- em geral, as heurísticas admissíveis tendem a ser consistentes

desempenho A*

- A* é otimamente eficiente

  nenhum outro algoritmo garante expandir menos nós

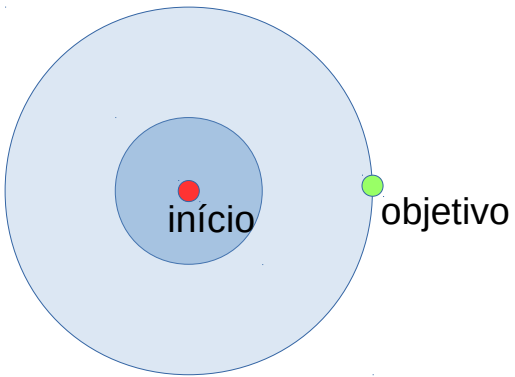
  hélas!…

- em muitos problemas o nº de nós a expandir é exponencial
  - $O(b^{\epsilon d}) = O((b^{\epsilon})^{d})$

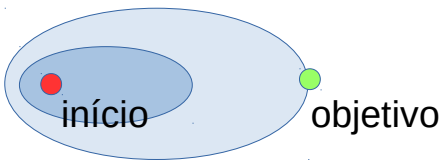
fator de ramificação efetivo: b^{ϵ}

em que ϵ é o erro relativo da heurística $\epsilon \equiv (h^* - h) / h^*$

A* vs. custo uniforme



custo uniforme
procura igualmente
em todas as direções



A*
procura tendencialmente
na direção do objetivo,
mas com contenção para
garantir otimalidade

aplicações do A*

- videojogos
- encaminhamento / navegação
- planeamento de recursos
- planeamento de movimentos de robôs
- análise de linguagem
- tradução automática
- reconhecimento de fala
- … (virtualmente qualquer problema de procura em grafo...)