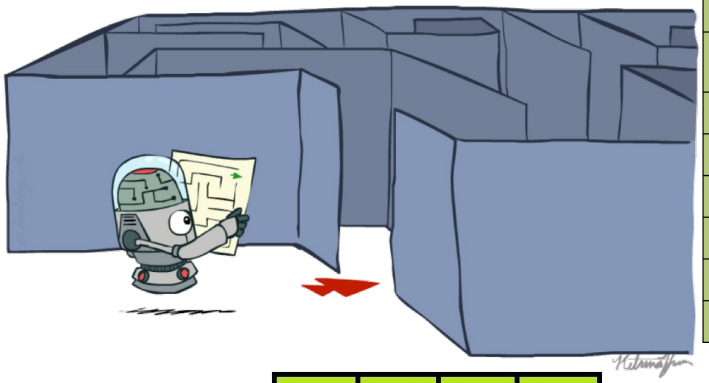
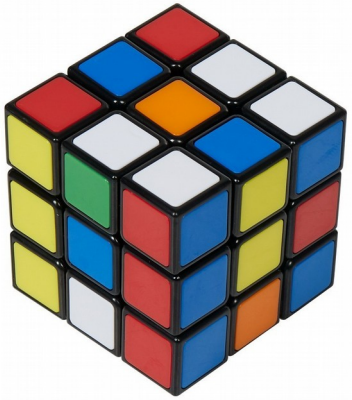
problemas de procura

problemas de procura



Horas	Segunda	Terça	Quarta	Quinta	Sexta		Sábado
08:00 - 08:30	[3LEI02] [1 2 21] [TP] TP12						
08:30 - 09:00							
09:00 - 09:30							
09:30 - 10:00	[3LMAAF01; 3LEI01; 3LEI02; 3LEI03; 3LEI04; 3LEI05] [3 2 15] [T] T11				[3LMAAF01; 3LEI01; 3LEI02; 3LEI03; 3LEI04; 3LEI05] [3 2 15] [T] T11		
10:00 - 10:30							
10:30 - 11:00							
11:00 - 11:30							
11:30 - 12:00	[3LEI04] [1 2 22] [TP] TP14		[3LEI05] [1 2 22] [TP] TP15		[3LMAAF01; 3LEI01] [1 2 24] [TP] TP11	[3LEI01] [1 2 26] [TP] TP16	
12:00 - 12:30				[3LEI03] [1 2 15] [TP] TP13			
12:30 - 13:00							
13:00 - 13:30							

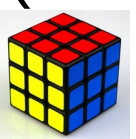
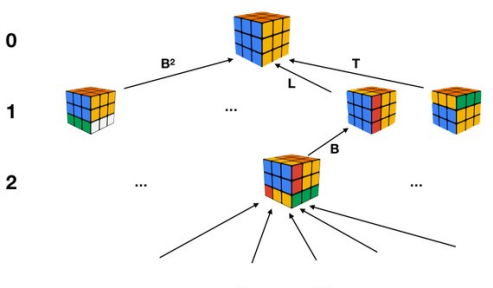


1	2	8	11
9	3	6	14
4	7	12	10
5	15	13	

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Ciências ULisboa

problema de procura

- estado inicial 
- muda de estado com uma ação 
- objetivo (estado final) 
- espaço de estados 

problemas de procura – tipos

- qual a sequência de ações para ir do estado inicial ao objetivo?
  - qual a melhor dessas sequências (menor custo)?

procura de plano de ação

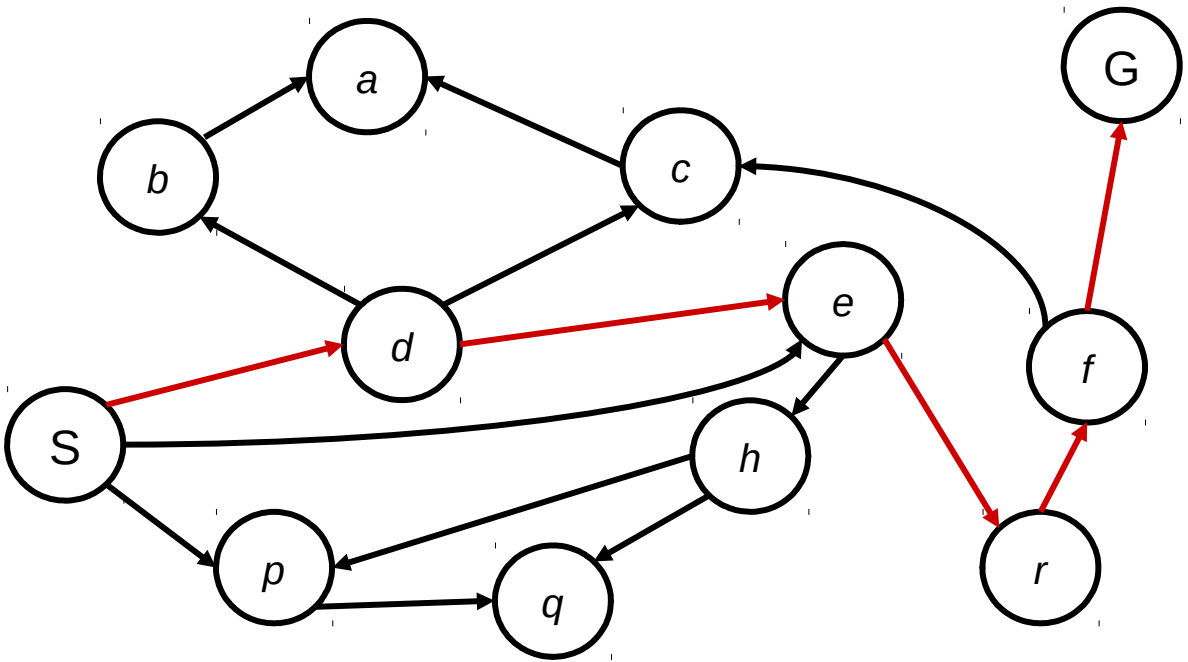
- qual uma solução para o problema?
  - não é necessária uma sequência de ações

procura com restrições

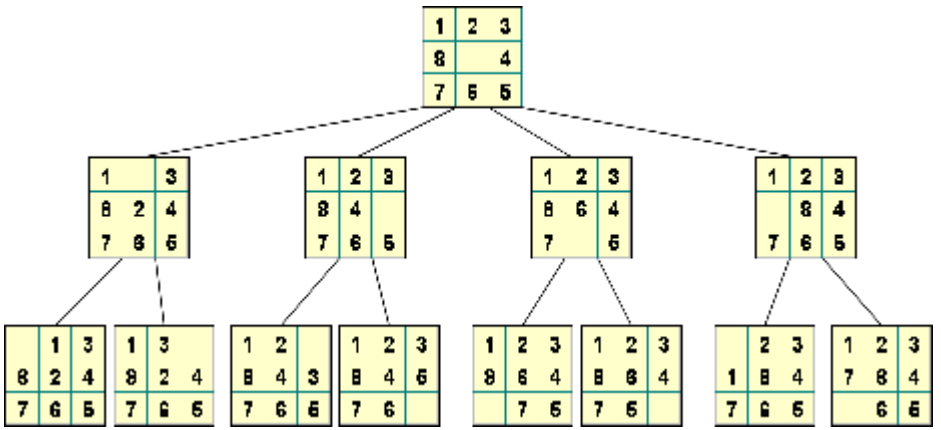
- qual a melhor ação neste estado?
  - para contrariar competição

procura com adversários

grafo do espaço de estados



árvore de procura



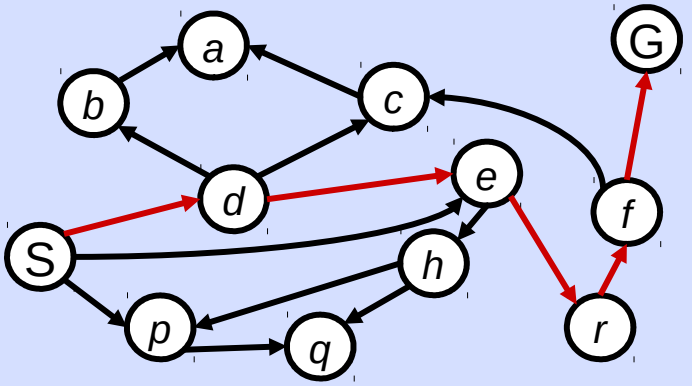
- nós = estados
- arcos = ações
- teste do objetivo → V | F

  pode haver mais do que 1 estado que atinge o objetivo (ver probl. rainhas)

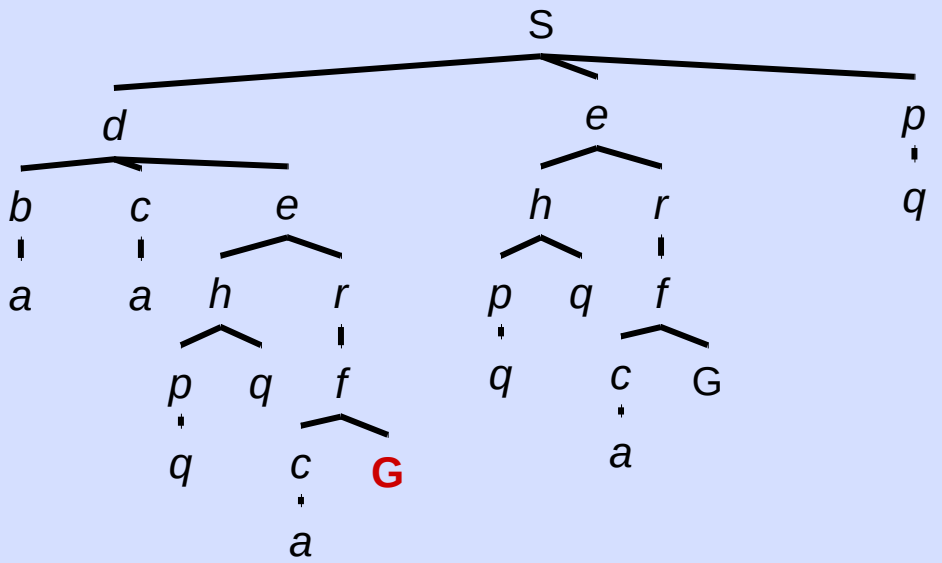
estrutura de dados não é normalmente construída na totalidade

grafo de espaço de estados vs. árvore de procura

Grafo de Espaço de Estados



Árvore de Procura



definição funcional

$s_{\text{início}}$: estado inicial

Actions(s): ações possíveis no estado s

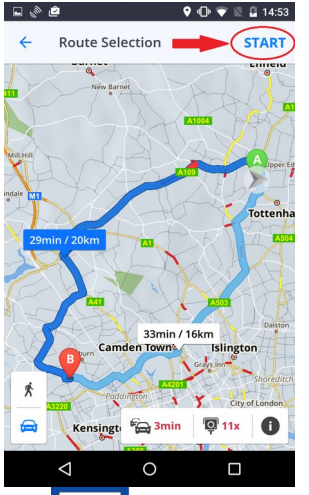
Cost(s, a): custo da ação a no estado s # caso definido

Succ(s, a): estado seguinte após ação a no estado s

IsEnd(s): teste de objetivo

- solução: sequência de ações que minimiza o custo

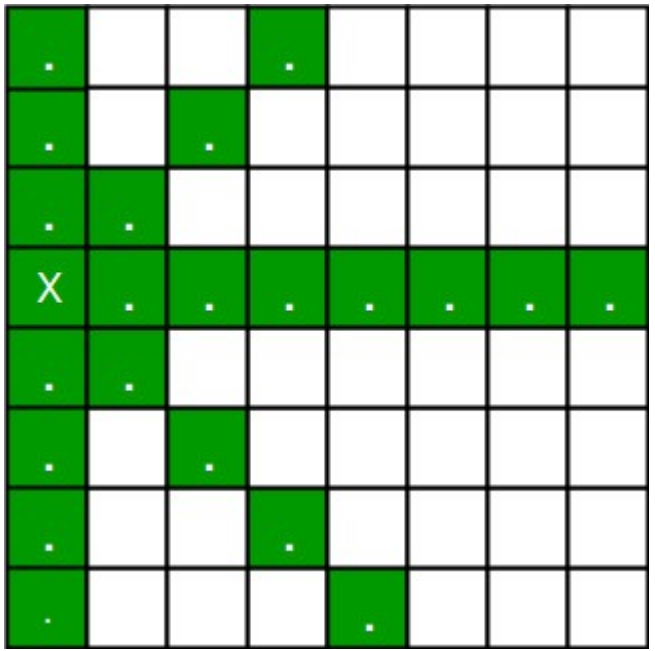
procura de plano de ação



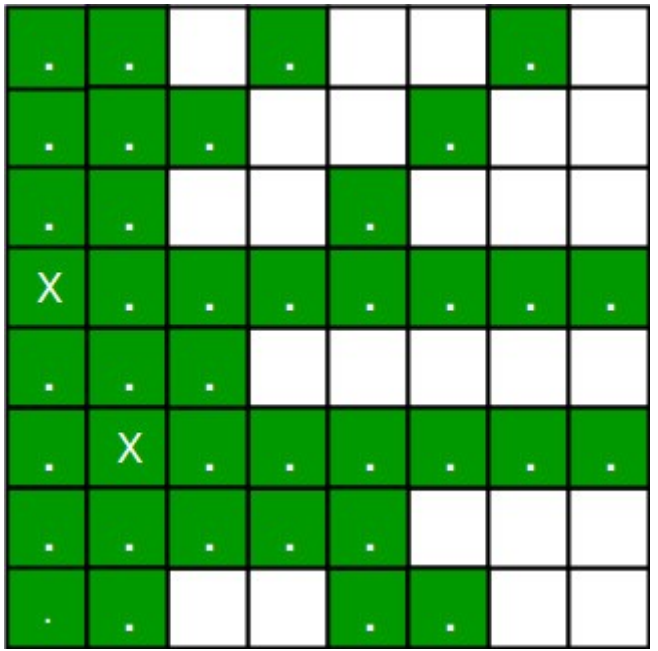
Ciências ULisboa

problema das 8 rainhas

- colocar 8 rainhas num tabuleiro de xadrez sem se atacarem



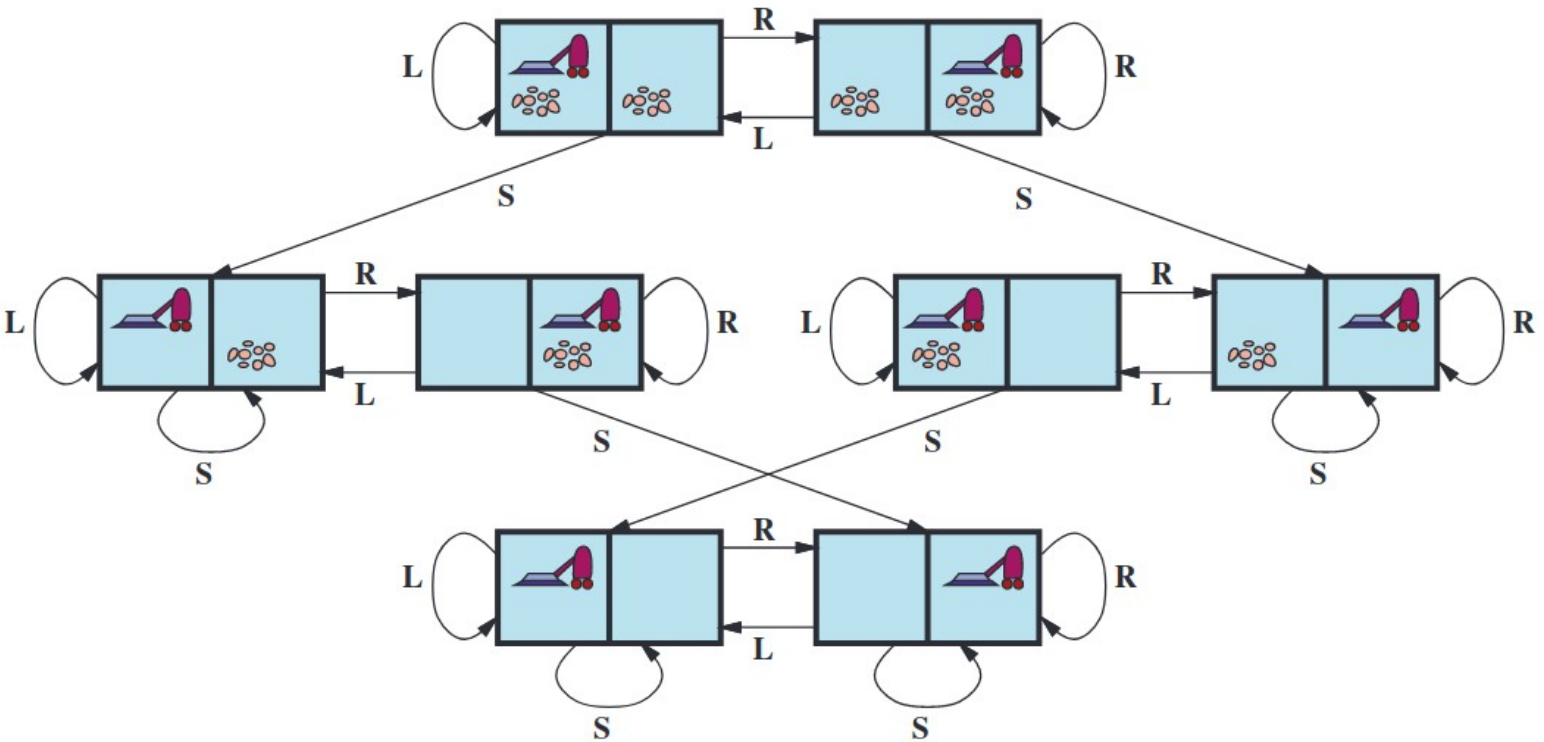
procura com restrições



jogo

procura com adversários

um problema simplificado *(toy)* – robô aspirador



é viável representar espaço de estados completo

Ciências ULisboa

robô aspirador - formulação

estado: localização do agente, localizações de lixo

com n localizações → $n.2^n$ estados

estado inicial: <qualquer um dos estados>

ações: *esquerda (L), direita (R), aspira (S)*

se ambiente 2D, deverá haver também *acima* e *abaixo*

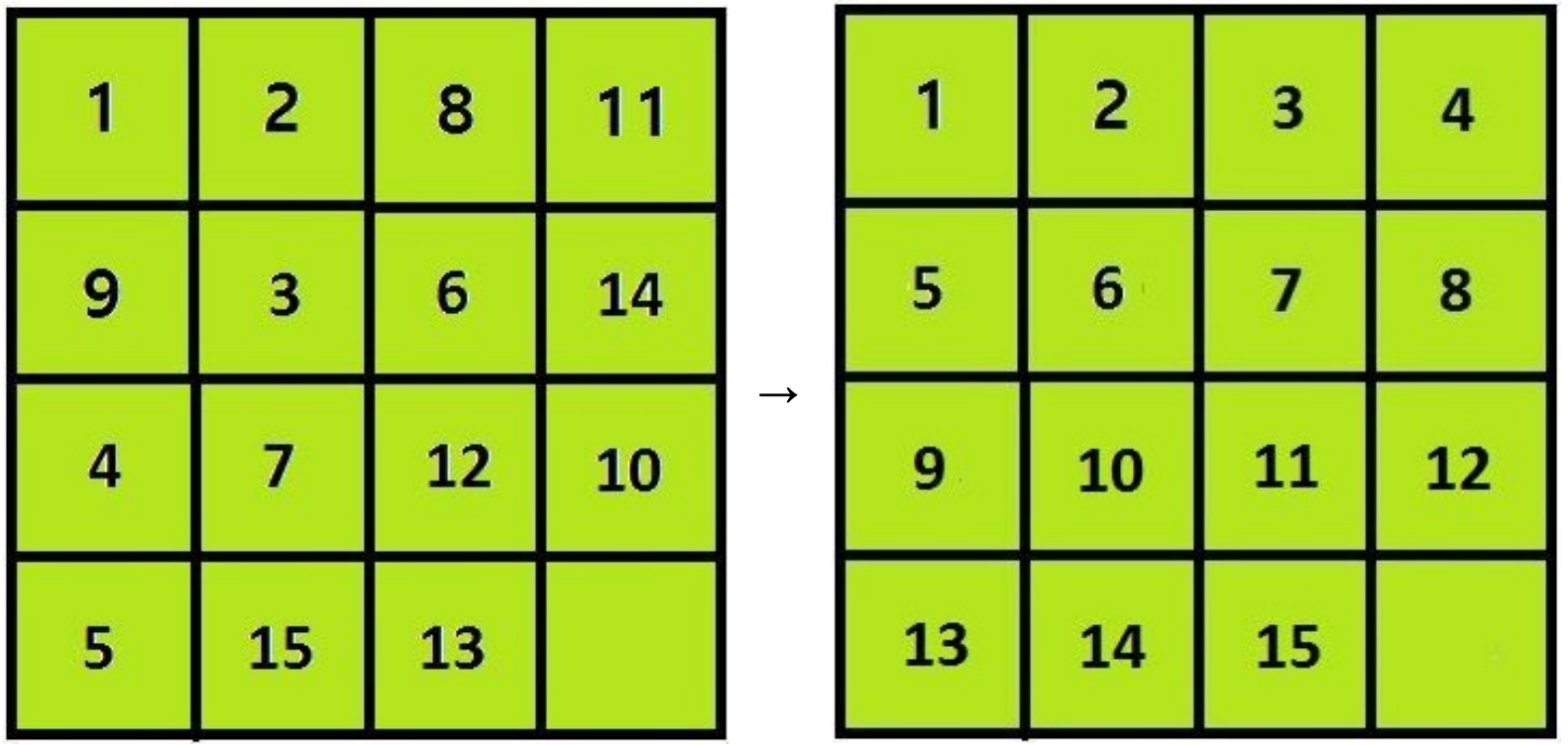
resultados das ações (`Succ`): determinísticos

não mexe se tenta ir para lá dos limites; aspirar limpo s/ efeito

teste do objetivo: localizações todas limpas

custo da sequência: nº de passos da solução

puzzle



puzzle - formulação

estado: localização de cada nº e do vazio

estado inicial: <qualquer um dos estados>

ações: movimento do vazio - *esquerda, direita, acima, abaixo*

é a formulação mais simples!

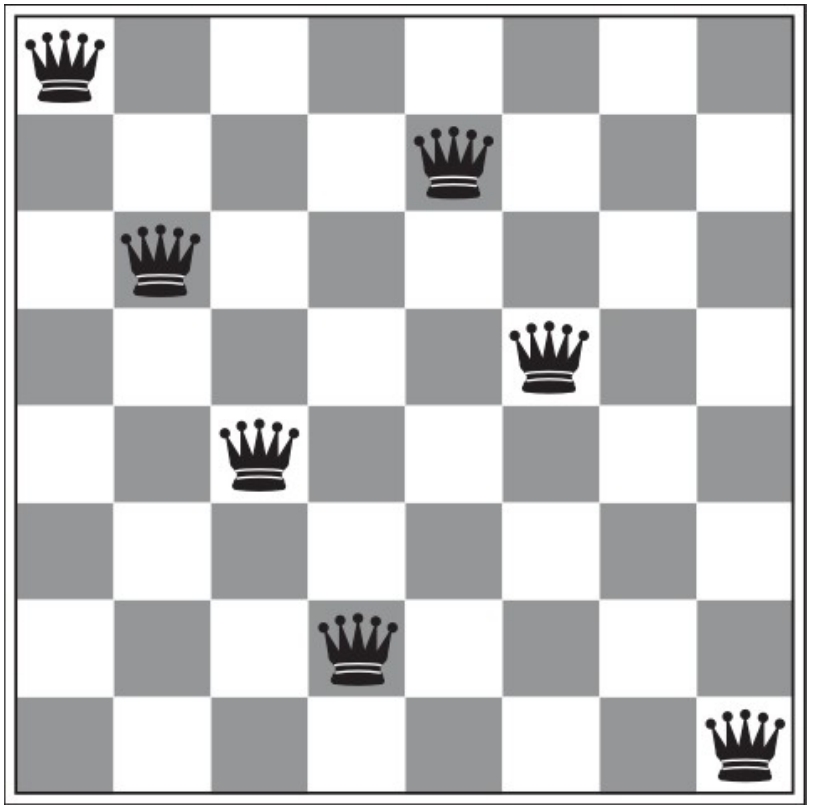
resultados das ações (`Succ`): determinísticos

troca a posição do vazio com o nº afetado pelo movimento

teste do objetivo: localizações dos nºs e vazio correspondem à configuração do objetivo

custo da sequência: nº de passos da solução

8 rainhas



procura com restrições

só interessa uma solução
(colocar as 8 rainhas)

não há sequência de ações no mundo

sim, é uma solução incompleta!
falta 1 rainha e há duas que se atacam

8 rainhas – formulação incremental

estado: qualquer colocação de 0 a 8 rainhas no tabuleiro

estado inicial: tabuleiro vazio

ações: colocar uma rainha numa casa vazia

resultados das ações (`Succ`): determinísticos

tabuleiro com mais uma rainha colocada

teste do objetivo: 8 rainhas colocadas, sem se atacarem

- sequências possíveis: 64 x 63 x … x 57 ≈ $1{,}8 \times 10^{14}$

8 rainhas – formulação incremental com restrições

- colocar uma rainha apenas numa casa não atacada

  estado: qualquer colocação de 0 a 8 rainhas, uma por coluna, nas colunas mais à esquerda, sem ataques (organização por colunas)

  ações: colocar uma rainha na coluna livre mais à esquerda, sem ser atacada pelas rainhas já no tabuleiro

- espaço de estados de dimensão 2057

caminho num mapa – um problema real

por (auto-)estrada

mapa como grafo

nó = povoação | cruzamento

arco = segmento de estrada

Ciências ULisboa

caminho num mapa - formulação

estado: percurso desde o estado inicial e respetivo custo

percurso = sequência de nós e arcos

estado inicial: percurso só com localização inicial e custo = 0

ações: arco escolhido em cada nó do percurso

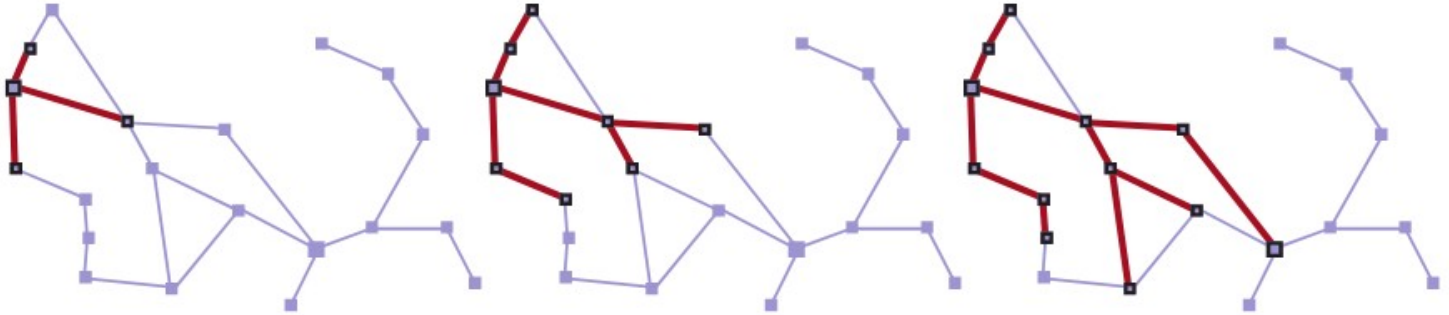
resultados das ações (`Succ`): determinísticos

acrescenta ao percurso um arco e nó final desse arco

teste do objetivo: nó final do percurso é o destino?

custo da sequência: distância | tempo | gastos (€)...

algoritmo de procura



1) formar uma árvore (grafo) de procura com o estado inicial como único nó = raiz da árvore – fronteira inicial
2) expandir um ou mais nós na fronteira

    aplicar todas as ações possíveis em cada nó a expandir fronteira e determinar os nós seguintes em cada caso

    acrescentar os novos nós à fronteira

3) repetir 2) até atingir objetivo, ou ter expandido todo o grafo

algoritmo geral

- todos os algoritmos de procura têm esta estrutura base

- diferenciam-se pela

  escolha dos nós a expandir – estratégia de procura

  (aulas seguintes)

estruturas de dados da procura

- árvore de procura
  - nó: estado correspondente no espaço de estados
    custo $g(n)$ desde a raiz,
    nó ascendente,
    ação desde o nó ascendente
- fronteira: fila de espera
- conjunto de nós expandidos: tabela de dispersão

  para evitar ciclos na procura

desempenho do algoritmo de procura

completude

garantia de encontrar uma solução, caso exista

otimalidade

garantia de encontrar a solução ótima

ótimo <=> custo mínimo

complexidade temporal

tempo que o algoritmo de procura demora a encontrar a solução

complexidade espacial

memória necessária para o algoritmo de procura encontrar a solução

complexidade da procura

b – fator de ramificação

d – profundidade do objetivo mais próximo da raiz

m – comprimento máximo de um caminho no espaço de estados

temporal ~ total de nós expandidos

espacial ~ nº máximo de nós em memória

custos

pode ser necessário usar

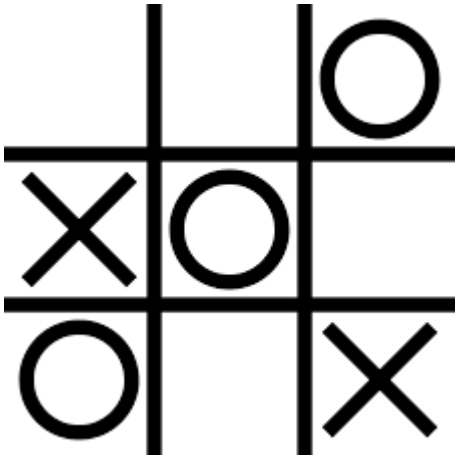
custo total = custo (tempo) da procura +
custo da solução

para problemas em tempo real

ex: navegação durante a viagem

necessário converter um dos custos na unidade do outro

problemas de treino de procura – formular espaço de estados



dar troco com moedas de
50, 20, 10, 5, 2 e 1 cêntimos
- supor que há número ilimitado de cada moeda

nota: assumir que após cada jogada do agente o mundo muda (por ação do oponente)