

SICO7A

SISTEMAS INTELIGENTES 1

Aula 03 A - Estratégias
para Busca

Prof. Rafael G. Mantovani

Roteiro

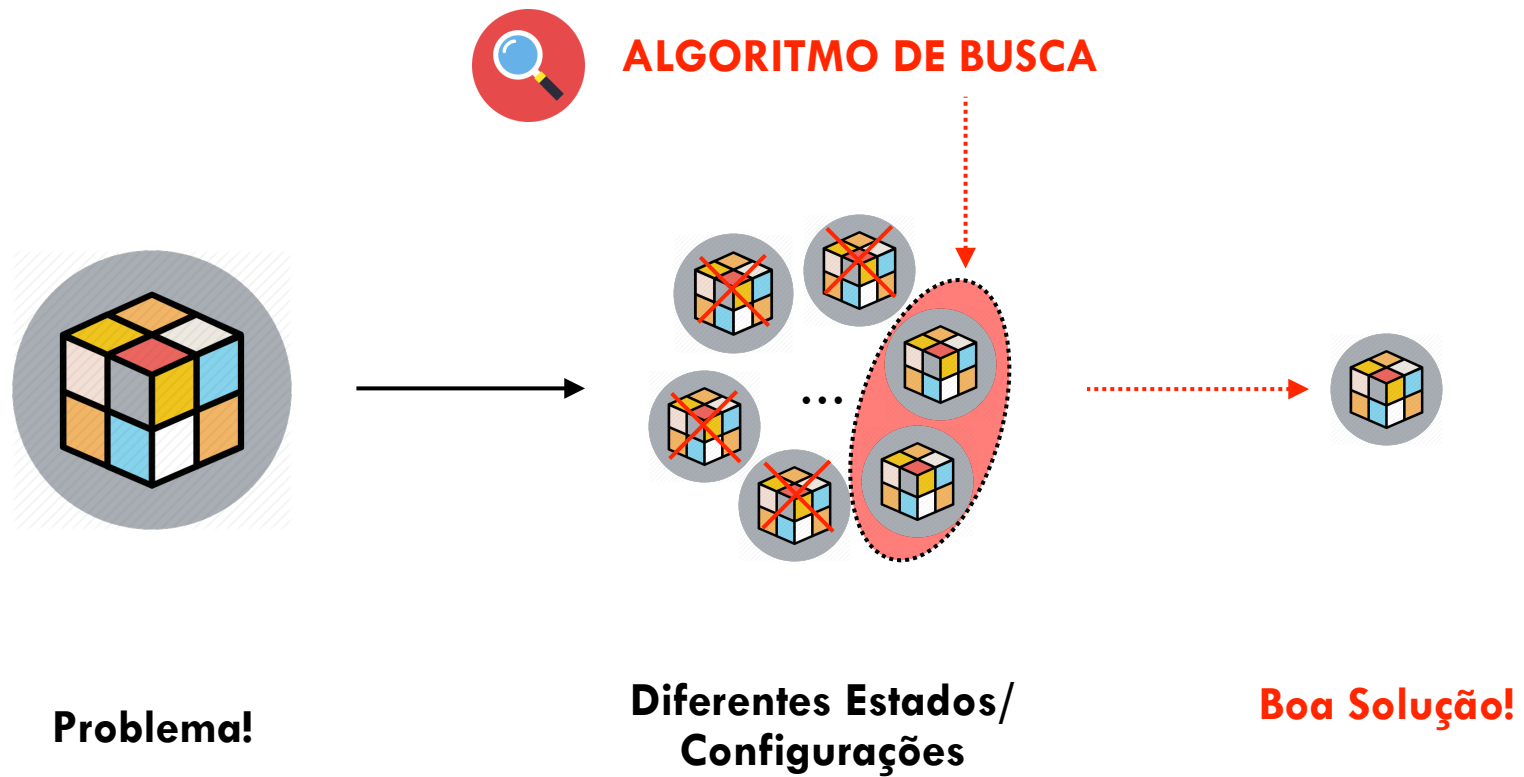


- 1** Introdução
- 2** Busca com Retrocesso (*Backtracking*)
- 3** Exemplo
- 4** Exercício
- 5** Referências

Roteiro

- 1** Introdução
- 2** Busca com Retrocesso (*Backtracking*)
- 3** Exemplo
- 4** Exercício
- 5** Referências

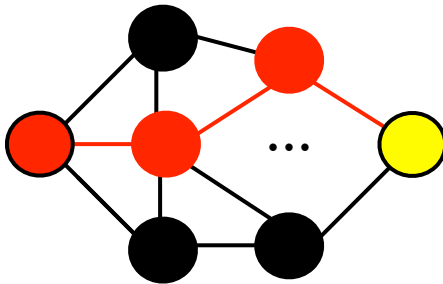
Introdução



Representação



ALGORITMO DE BUSCA



**Grafo de Espaço
de Estados**

- estado / solução parcial
- estado(s) inicial(ais)
- estado(s) final(ais) / objetivos
- caminho de solução

Representação



ALG. DE BUSCA



Algum percurso

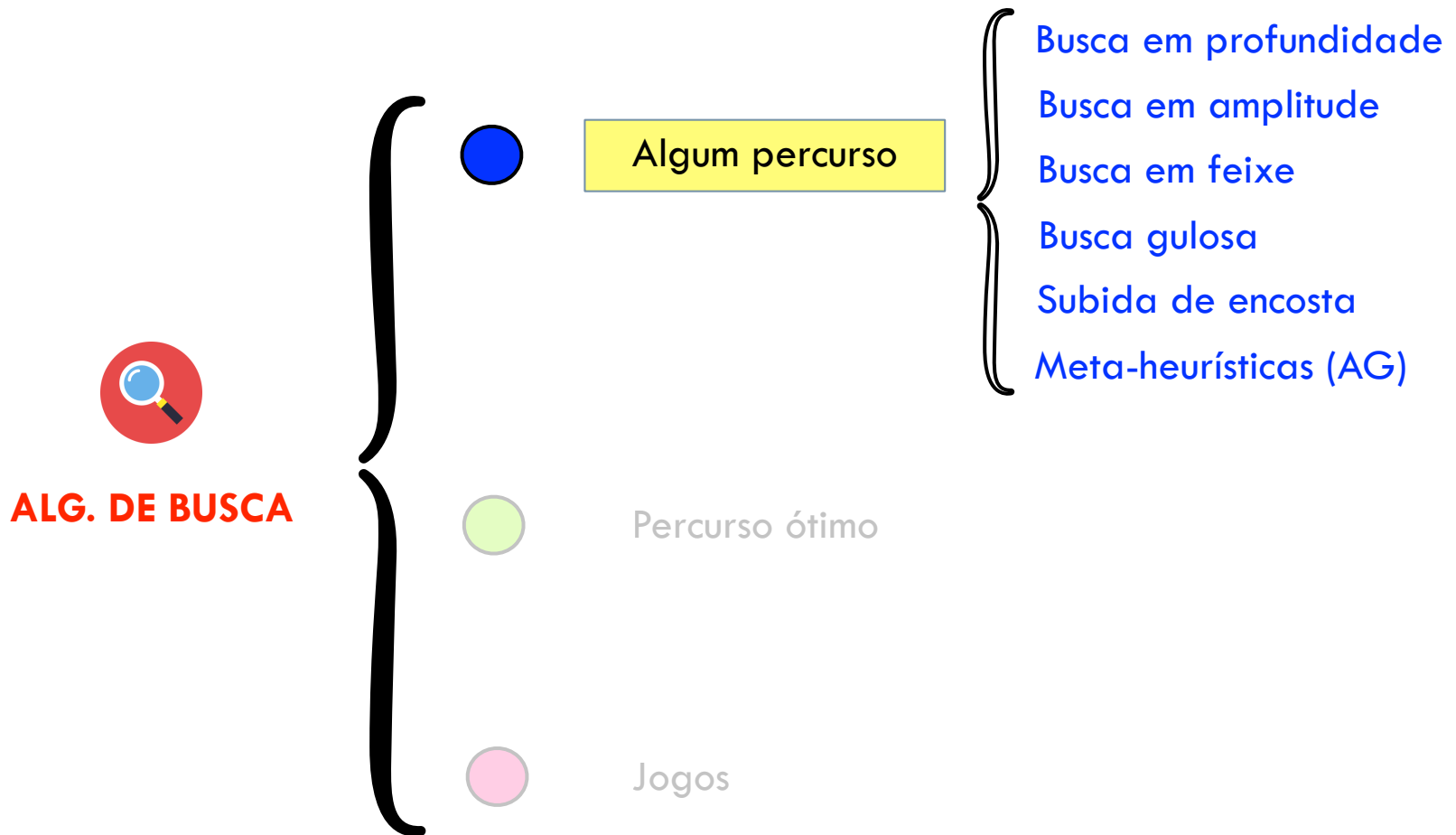


Percurso ótimo

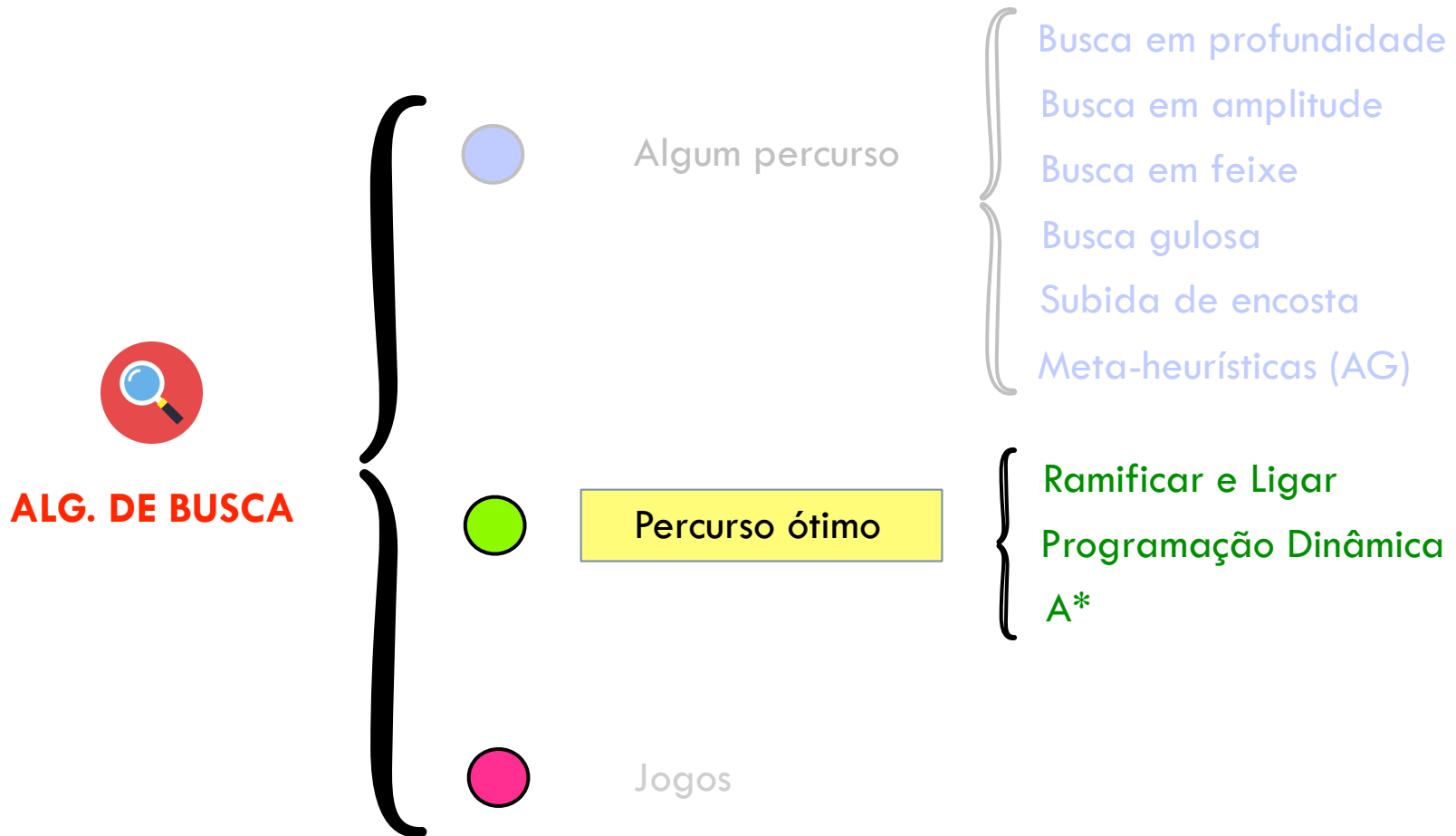


Jogos

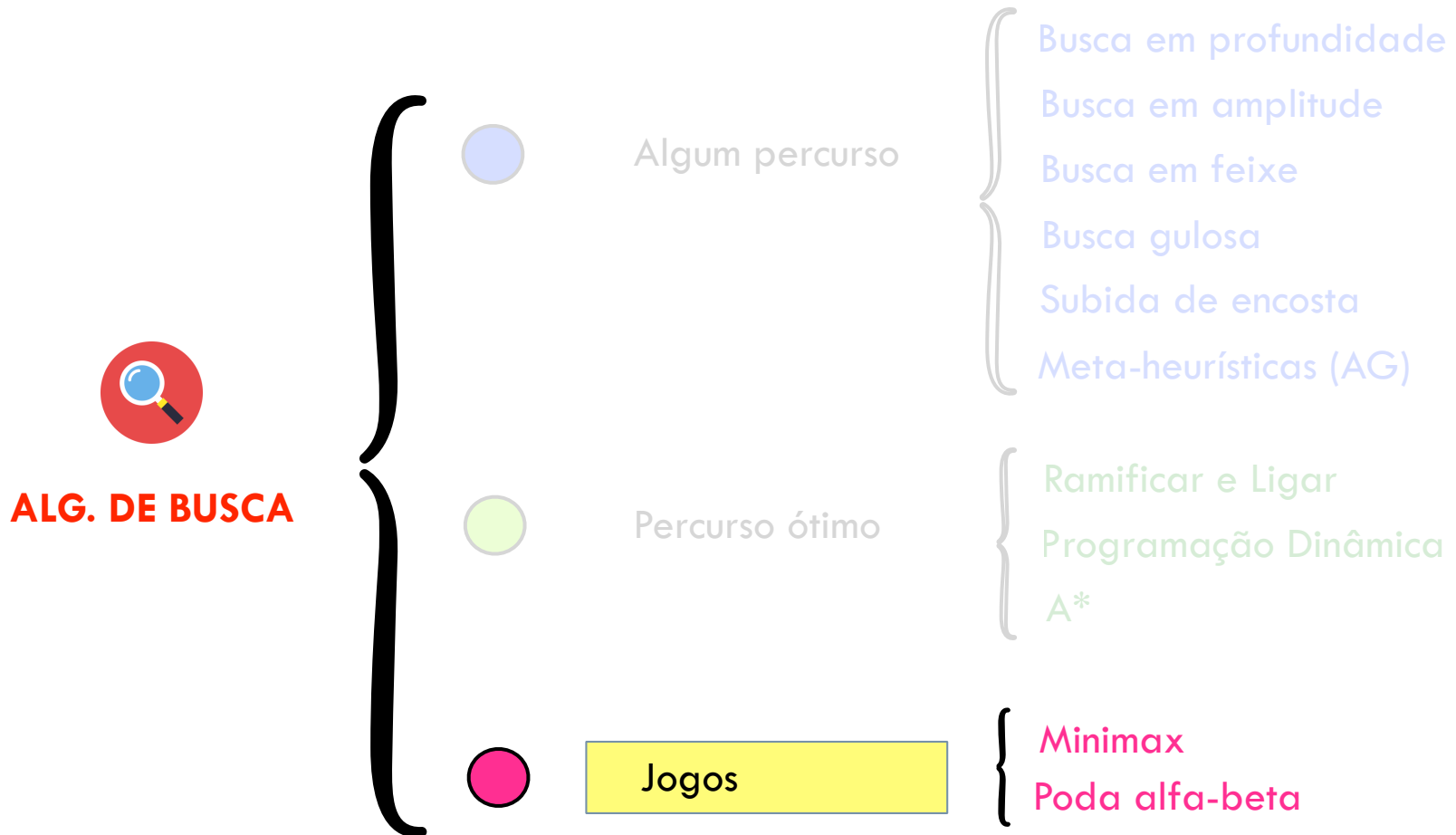
Representação



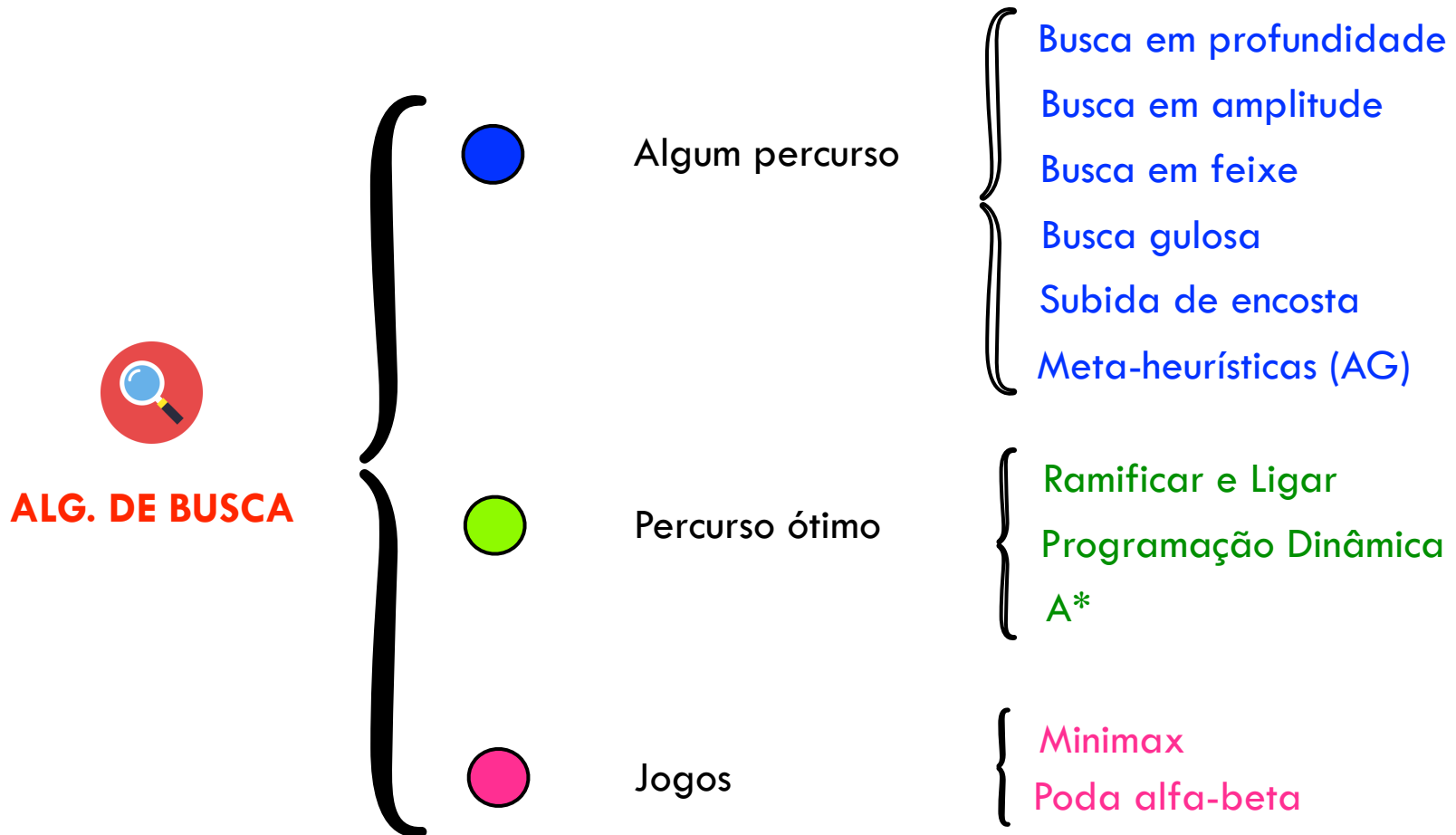
Representação



Representação



Representação

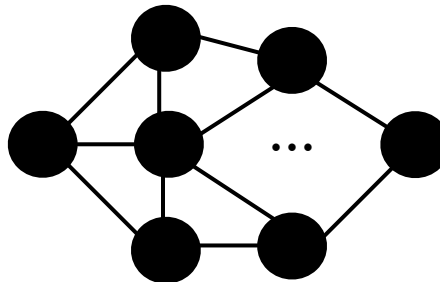


Roteiro

- 1 Introdução
- 2 Busca com Retrocesso (*Backtracking*)
- 3 Exemplo
- 4 Exercício
- 5 Referências

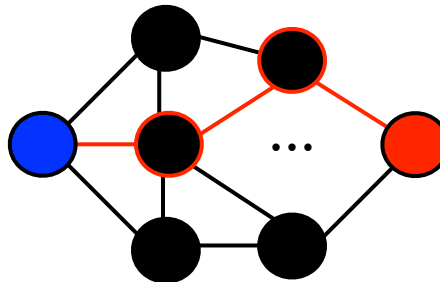
Busca com retrocesso (*backtracking*)

* **resolvedor** deve achar um caminho de um estado inicial até um estado objetivo através do grafo do espaço de estados.



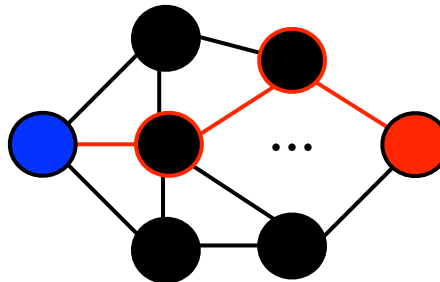
Busca com retrocesso (*backtracking*)

* **resolvedor** deve achar um caminho de um estado inicial até um estado objetivo através do grafo do espaço de estados.



Busca com retrocesso (*backtracking*)

* **resolvedor** deve achar um caminho de um estado inicial até um estado objetivo através do grafo do espaço de estados.



Se achou objetivo: retorna sucesso :)
Se chegou em um beco sem saída: procurar outro caminho possível

Algoritmo: *Backtracking*



Algoritmo: *Backtracking*

- 1 **SE** o estado atual S atender os requisitos da descrição do objetivo, **ENTÃO:**
gere seu primeiro descendente S_{filho1} e aplique o procedimento *Backtracking* recursivamente a este nó

Algoritmo: *Backtracking*

- 1 **SE** o estado atual S atender os requisitos da descrição do objetivo, **ENTÃO**:
gere seu primeiro descendente S_{filho1} e aplique o procedimento *Backtracking* recursivamente a este nó
- 2 **SE** o *Backtracking* não achar um nó objetivo no subgrafo radicado em S_{filho1} , repita o procedimento para o seu irmão S_{filho2}

Algoritmo: *Backtracking*

- 1 **SE** o estado atual S atender os requisitos da descrição do objetivo, **ENTÃO**:
gere seu primeiro descendente S_{filho1} e aplique o procedimento *Backtracking* recursivamente a este nó
- 2 **SE** o *Backtracking* não achar um nó objetivo no subgrafo radicado em S_{filho1} , repita o procedimento para o seu irmão S_{filho2}
- 3 **Continue** até que algum descendente de um filho seja um nó objetivo, ou até que todos os filhos tenham sido buscados

Algoritmo: *Backtracking*

- 1 **SE** o estado atual S atender os requisitos da descrição do objetivo, **ENTÃO**:
gere seu primeiro descendente S_{filho1} e aplique o procedimento *Backtracking* recursivamente a este nó
- 2 **SE** o *Backtracking* não achar um nó objetivo no subgrafo radicado em S_{filho1} , repita o procedimento para o seu irmão S_{filho2}
- 3 **Continue** até que algum descendente de um filho seja um nó objetivo, ou até que todos os filhos tenham sido buscados
- 4 **SE** nenhum dos filhos de S levar a um objetivo, então o algoritmo “falha” para o nó inicial S .

Algoritmo: *Backtracking*

Características Gerais:

- ★ Algoritmo continua até encontrar um objetivo ou esgotar o espaço de busca
- ★ Usa retrocesso (backtracking) para testar outras possibilidades de caminhos que não foram avaliados anteriormente

Algoritmo: *Backtracking*

Sugestão de implementação:

■ EC:

■ LE:

■ LNE:

■ BSS:

Algoritmo: *Backtracking*

Sugestão de implementação:

- **EC:** *estado corrente*
- **LE:** *lista de estados* - lista os estados no caminho atual
- **LNE:** *lista de novos estados* - nós que aguardam avaliação, e cujos descendentes não foram gerados e buscados
- **BSS:** *beco sem saída* - estados cujos descendentes não contém objetivo.

Algoritmo: *Backtracking*

Sugestão de implementação:

- **EC:** *estado corrente*
- **LE:** *lista de estados* - lista os estados no caminho atual
- **LNE:** *lista de novos estados* - nós que aguardam avaliação, e cujos descendentes não foram gerados e buscados
- **BSS:** *beco sem saída* - estados cujos descendentes não contém objetivo.

Se um estado recém gerado pertence a alguma das listas, então ele já foi visitado e é **ignorado**.

Pseudocódigo



BUSCA_RETROCESSO (G, Inicial):

Pseudocódigo

BUSCA_RETROCESSO (G, Inicial):

```
1.  LE = [Inicial]; LNE = [Inicial]; BSS = [ ]; // Inicialização
2.  EC = Inicial;                               // Estado corrente
3.  Enquanto LNE != [ ], faça:
4.  | Se EC == Objetivo (ou atende descrição do objetivo), então:
5.  | | Retorna LE; // Se houver sucesso, retorna a lista de estados no caminho

6.  | Se EC não tem filhos, então:
7.  | | Enquanto LE não está vazio e EC == o primeiro elemento de LE,
8.  | | faça:
9.  | | | acrescenta EC em BSS;
10. | | | remove primeiro elemento de LE;
11. | | | remove primeiro elemento de LNE;
12. | | | EC = primeiro elemento de LNE;
```

Pseudocódigo

BUSCA_RETROCESSO (G, Inicial):

```

13.         acrescenta EC a LE;
14.     Senão:
15.         coloca os filhos de EC em LNE; // Exceto os já em BSS, LE ou LNE
16.         EC = primeiro elemento de LNE;
17.         acrescenta EC a LE;
18.     Retorna FALHA;

```

Pseudocódigo

BUSCA_RETROCESSO (G, Inicial):

```

13.         acrescenta EC a LE;
14.     Senão:
15.         coloca os filhos de EC em LNE; // Exceto os já em BSS, LE ou LNE
16.         EC = primeiro elemento de LNE;
17.         acrescenta EC a LE;
18.     Retorna FALHA;

```

- * **EC** é o estado corrente
- * Operadores são aplicados a EC, e os **filhos ordenados** por algum critério
- * Se EC não tiver filhos, ele é removido de LE (**backtracking**)

Roteiro



- 1 Introdução
- 2 Busca com Retrocesso (*Backtracking*)
- 3 Exemplo
- 4 Exercício
- 5 Referências

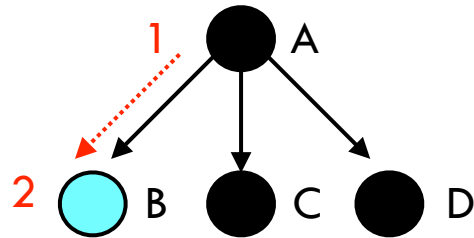
Exemplo



1 ○ A

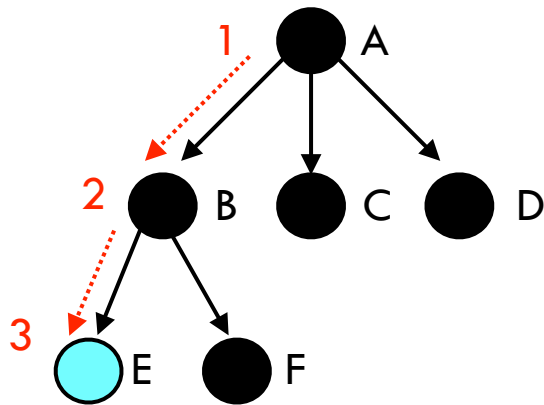
It	EC	LE	LNE	BSS
0	A	[A]	[A]	[]

Exemplo



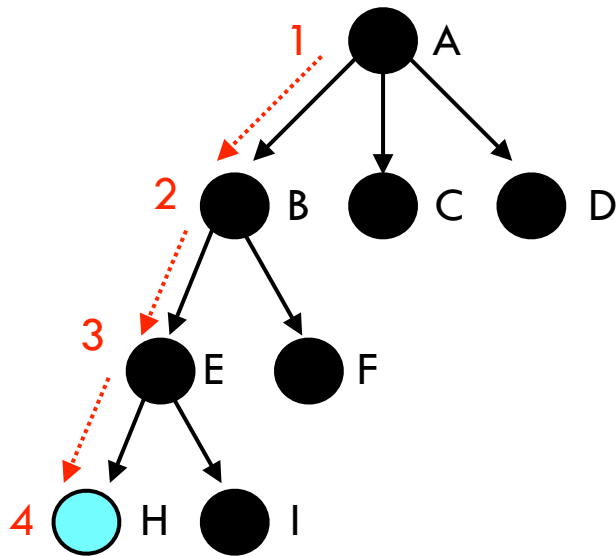
It	EC	LE	LNE	BSS
0	A	[A]	[A]	[]
1	B	[BA]	[BCDA]	[]

Exemplo



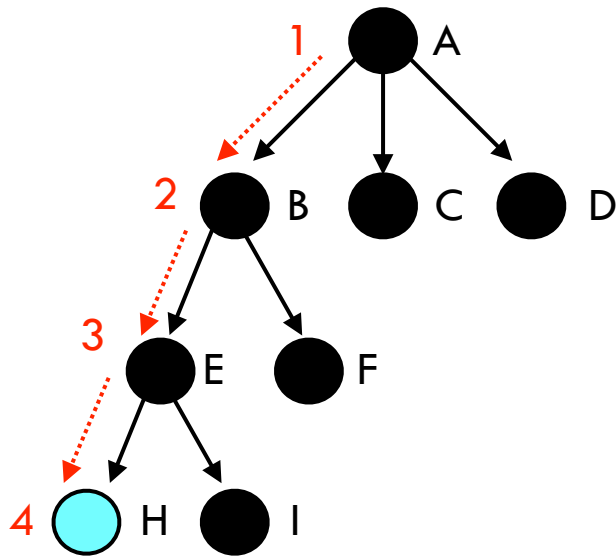
It	EC	LE	LNE	BSS
0	A	[A]	[A]	[]
1	B	[BA]	[BCDA]	[]
2	E	[EBA]	[EFBCDA]	[]

Exemplo



It	EC	LE	LINE	BSS
0	A	[A]	[A]	[]
1	B	[BA]	[BCDA]	[]
2	E	[EBA]	[EFBCDA]	[]
3	H	[HEBA]	[HIEFBCDA]	[]

Exemplo



It	EC	LE	LNE	BSS
0	A	[A]	[A]	[]
1	B	[BA]	[BCDA]	[]
2	E	[EBA]	[EFBCDA]	[]
3	H	[HEBA]	[HIEFBCDA]	[]

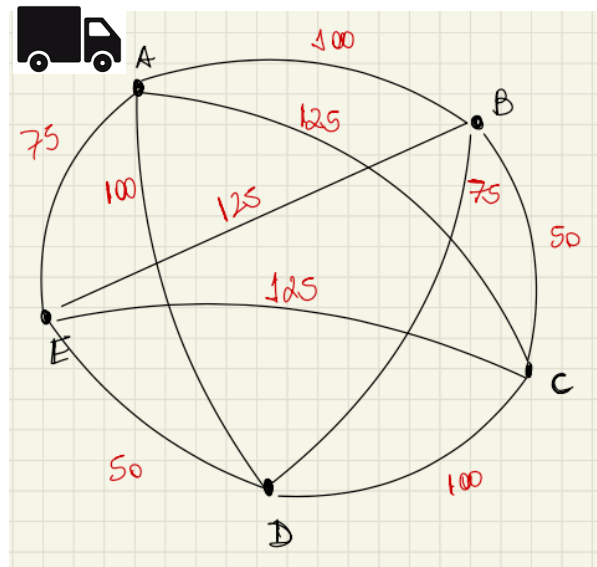
Continua ...

Roteiro

- 1 Introdução
- 2 Busca com Retrocesso (*Backtracking*)
- 3 Exemplo
- 4 Exercício
- 5 Referências

Exercícios

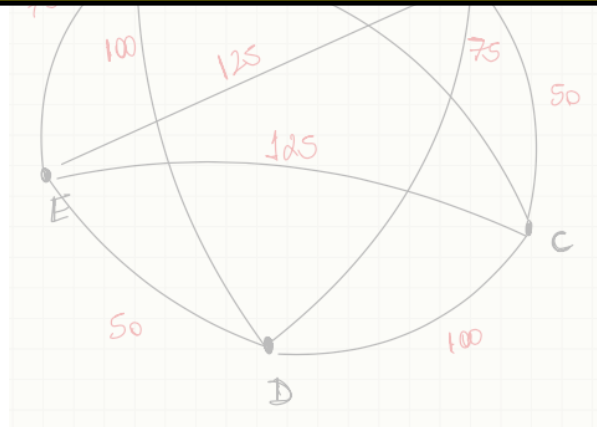
1) Reuna-se com seu grupo e implemente o algoritmo de busca com retrocesso em Python. O algoritmo será aplicado em instâncias de TSPs (*travelling salesman problems*). Logo, o algoritmo recebe um grafo e um vértice inicial, e retorna ou não um caminho possível.



Exercícios

1) Reuna-se com seu grupo e implemente o algoritmo de busca com retrocesso em Python. O algoritmo será aplicado em instâncias de TSPs (*travelling salesman problems*). Logo, o algoritmo recebe um grafo e um vértice inicial, e retorna ou não um caminho possível.

Objetivo: ter uma estrutura comum de algoritmo, a ser explorada nos demais algoritmos de busca.

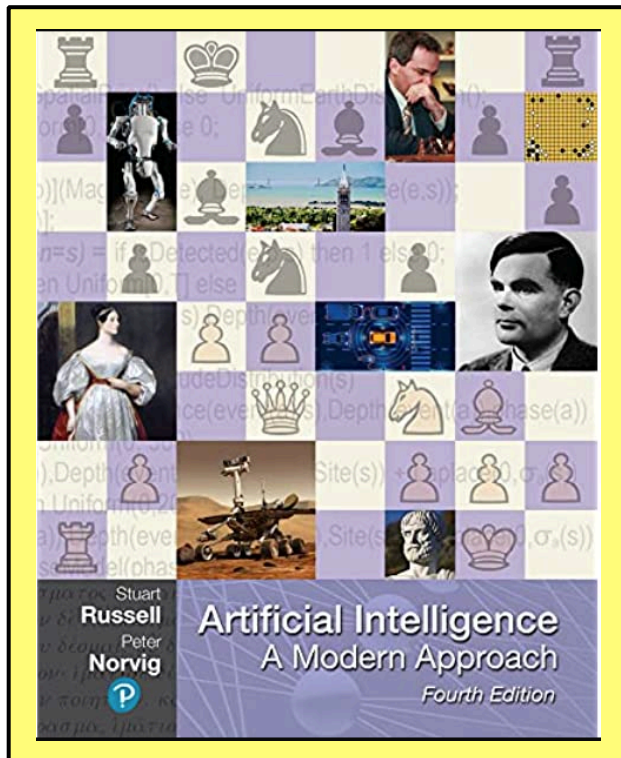


Roteiro

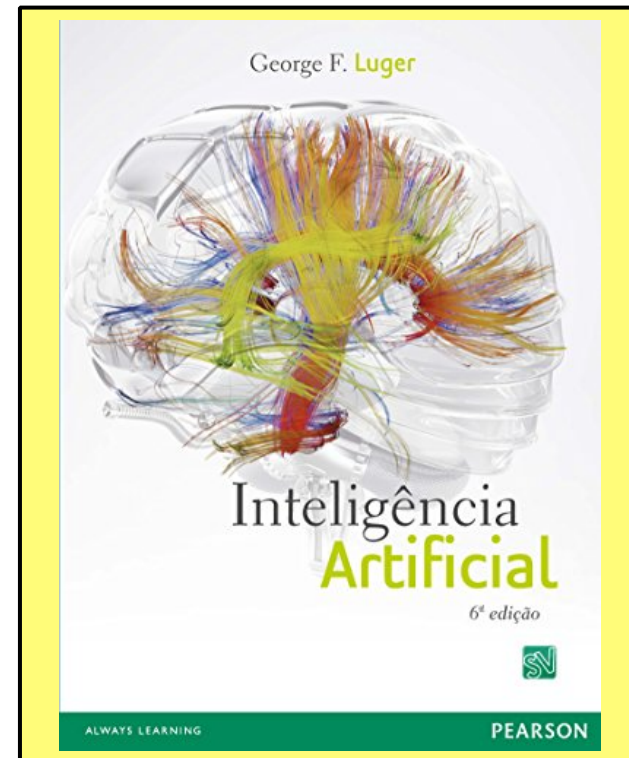


- 1 Introdução
- 2 Busca com Retrocesso (*Backtracking*)
- 3 Exemplo
- 4 Exercício
- 5 Referências

Referências sugeridas



[Russel & Norvig, 2021]

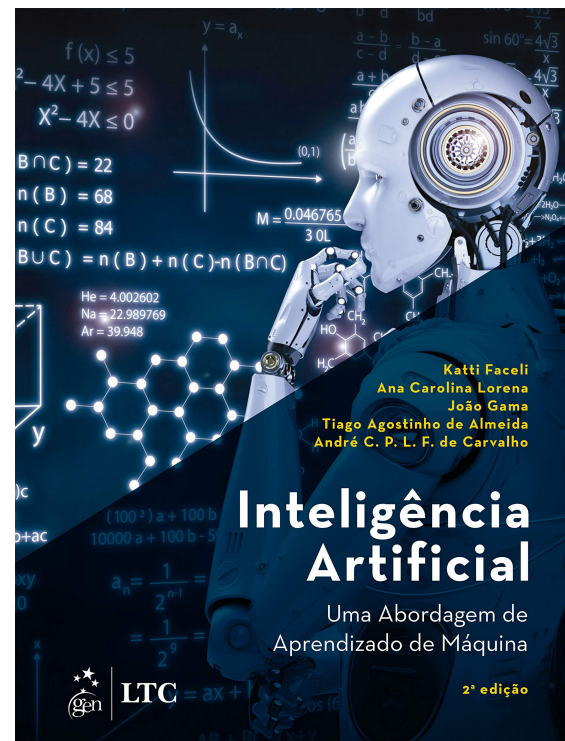


[Luger, 2013]

Referências sugeridas



[Coppin, 2010]



[Faceli et al, 2021]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br