

Inteligência Artificial

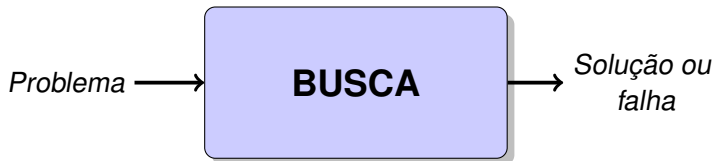
Profa. Dra. Viviane Menezes

vivianemenezes@ufc.br



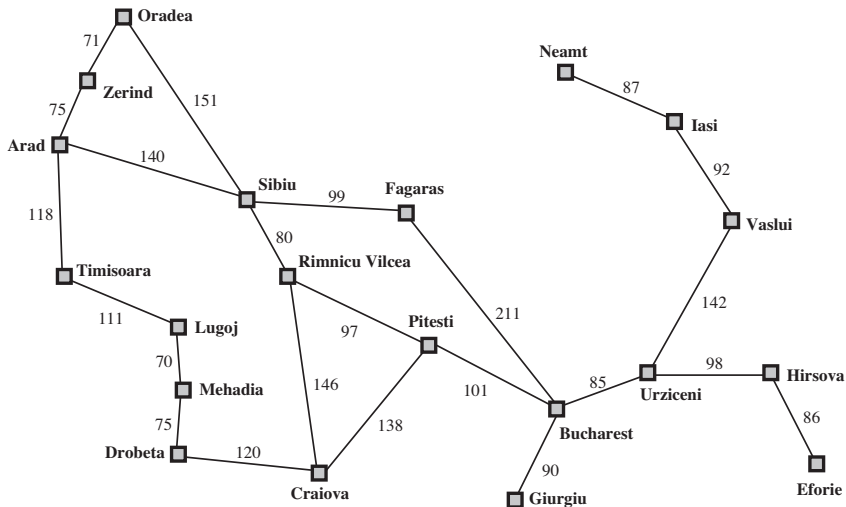
UNIVERSIDADE
FEDERAL DO CEARÁ
Campus Quixadá

Em Busca de uma Solução



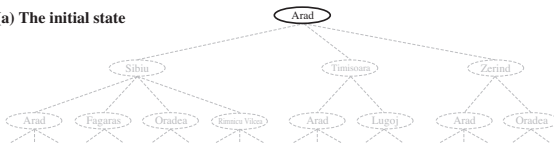
Construção de uma **árvore de busca**: os nós são formados pelos estados do problema; os ramos são as ações.

Em Busca de uma Solução

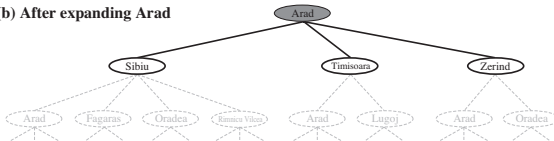


Em Busca de uma Solução

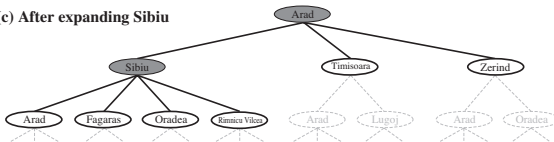
(a) The initial state



(b) After expanding Arad



(c) After expanding Sibiu



Em Busca de uma Solução

```
01. BUSCA(problema){
02.   iniciar borda com estado inicial
03.   iniciar explorados com vazio
03.   repita
04.     se borda está vazia
05.       retorne falha
06.     nó ← remover um nó da borda
07.     se nó contém um estado objetivo então
08.       retorne solução
09.     adicionar nó a explorados.
09.     expandir o nó se não estiver na borda ou em explorados.
10. }
```

- ***borda***: nós aguardando serem visitados.
- ***explorados***: **estados** já visitados.

Em Busca de uma Solução

Árvore de Busca: Estrutura Nó

- **Estado:** o estado a que o nó corresponde;
- **Pai:** o nó na árvore que gerou esse nó;
- **Ação:** a ação que foi aplicada ao pai para gerar o nó;
- **Custo do Caminho:** o custo de sair do estado inicial e alcançar o nó.

Em Busca de uma Solução

Algoritmos de Busca

- Busca em Largura
- Busca de Custo Uniforme
- Busca em Profundidade

As estratégias diferenciam-se na forma como a fila *da borda* é implementada.

Em Busca de uma Solução

Algoritmos de Busca

- Busca em Largura
- Busca de Custo Uniforme
- Busca em Profundidade

Busca em Largura

- Expande os nós, nível a nível.
- A **borda** é uma fila FIFO (*First In First Out*)

Busca em Largura

```
01. BUSCA-EM-LARGURA(problema){
02.   /* borda com nó.estado inicial e explorados vazio */
03.   repita
04.     se borda está vazia
05.       retorne falha
06.     nó ← remover elemento da borda
07.     adicionar nó.estado a explorados
08.     para cada ação aplicável em nó.estado
09.       filho ← criar nó filho
10.       se filho.estado não está em explorados ou borda
11.         se filho.estado é objetivo
12.           retorne solução
13.         adicionar filho em borda
13. }
```

Atividade

- Implementação da busca em largura seguindo o roteiro disponível no Moodle.

Em Busca de uma Solução

Algoritmos de Busca

- Busca em Largura
- Busca de Custo Uniforme
- Busca em Profundidade

Busca de Custo Uniforme

- Expande os nós nível a nível.
- Expande o nó com **menor custo de caminho**.
- A **borda** é uma fila de prioridades.

Busca de Custo Uniforme

```
01. BUSCA-DE-CUSTO-UNIFORME(problema){
02.   /* borda com nó.estado inicial e explorados vazio */
03.   repita
04.     se borda está vazia retorne falha
05.     nó ← remover elemento da borda
06.     se nó.estado é objetivo retorne solução
07.     adicionar nó.estado a explorados
08.     para cada ação aplicável em nó.estado
09.       filho ← criar nó filho
10.       se filho.estado não está em explorados ou borda
11.         adicionar filho em borda
12.       senão se filho.estado está na borda com maior custo
13.         substituir nó borda por filho
14.
15. }
```

Atividade

- Implementação da busca de custo uniforme seguindo o roteiro disponível no Moodle.