

Introdução à Inteligência Artificial

Parte 3

Evandro J.R. Silva

ejrs.profissional@gmail.com

Faculdade Estácio Teresina

30 e 31 de Janeiro



Estácio

Sumário

- 1 Resolução de Problemas por meio de Busca
 - Estratégias de Busca
- 2 Busca com Informação
 - Busca Gulosa
 - Busca A*
- 3 FIM



Resolução de Problemas por meio de Busca



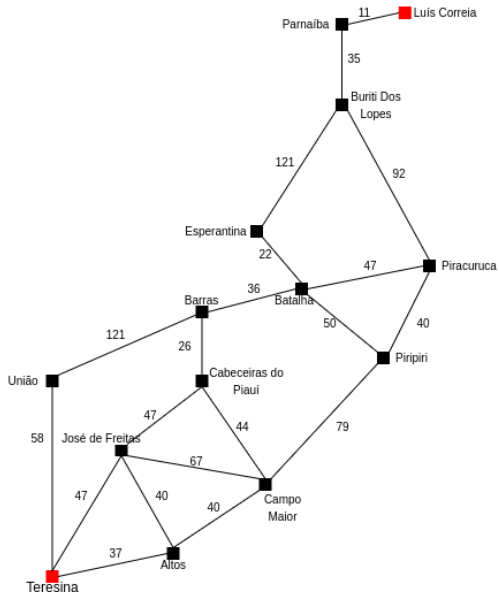
Estácio

Resolução de Problemas por meio de Busca

- Este é o caso de um agente baseado em objetivo.
- Vamos começar com o tipo mais simples de ambiente de tarefa, para o qual a solução para um problema é sempre uma sequência fixa de ações.



Resolução de Problemas por meio de Busca



Resolução de Problemas por meio de Busca

- **Problema:** ir de **Teresina** para **Luís Correia**.



Resolução de Problemas por meio de Busca

- **Problema:** ir de **Teresina** para **Luís Correia**.
- A partir de Teresina existem 3 possibilidades. Se o ambiente for desconhecido, ou seja, se o agente não tem qualquer informação adicional, terá de escolher entre uma das três aleatoriamente.



Resolução de Problemas por meio de Busca

- **Problema:** ir de **Teresina** para **Luís Correia**.
- A partir de Teresina existem 3 possibilidades. Se o ambiente for desconhecido, ou seja, se o agente não tem qualquer informação adicional, terá de escolher entre uma das três aleatoriamente.
- Mas vamos levar em consideração que o agente possui o mesmo mapa que vimos.
 - Portanto o ambiente é **observável** (pois o agente sempre conhecerá o estado atual), **discreto** (pois para cada estado existe um número finito de ações a escolher), **conhecido** (agora o agente sabe quais estados serão alcançados após uma ação) e **determinístico** (pois cada ação tem exatamente um resultado).



Resolução de Problemas por meio de Busca

■ Temos

- **Estado Inicial:** Está em Teresina.
- **Objetivo:** Estar em Luís Correia.
- **Ações:** Ir para alguma cidade.
- **Espaço de estados:** todos os estados acessíveis a partir do estado inicial.
- **Teste de objetivo:** Estamos em Luís Correia?



Resolução de Problemas por meio de Busca

■ Temos

- **Estado Inicial:** Está em Teresina.
 - **Objetivo:** Estar em Luís Correia.
 - **Ações:** Ir para alguma cidade.
 - **Espaço de estados:** todos os estados acessíveis a partir do estado inicial.
 - **Teste de objetivo:** Estamos em Luís Correia?
- Custo de caminho: quanto “custa” acrescentar um determinado estado? E como calcular esse custo?



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

- **Estados:** Uma descrição de estado especifica a posição de cada uma das oito peças e do quadrado vazio em um dos nove quadrados.



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

- **Estados:** Uma descrição de estado especifica a posição de cada uma das oito peças e do quadrado vazio em um dos nove quadrados.
- **Estado inicial:** O que está na figura, porém qualquer estado pode ser designado como o estado inicial.



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

- **Estados:** Uma descrição de estado especifica a posição de cada uma das oito peças e do quadrado vazio em um dos nove quadrados.
- **Estado inicial:** O que está na figura, porém qualquer estado pode ser designado como o estado inicial.
- **Ações:** A formulação mais simples define as ações como movimentos do quadrado vazio, *Esquerda, Direita, Para Cima ou Para Baixo.*



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

- **Estado inicial:** O que está na figura, porém qualquer estado pode ser designado como o estado inicial.
- **Ações:** A formulação mais simples define as ações como movimentos do quadrado vazio, *Esquerda*, *Direita*, *Para Cima* ou *Para Baixo*.
- **Modelo de Transição:** (não pode ser esquecido enquanto você estiver produzindo o programa de agente) Dado um estado e ação, ele devolve o estado resultante; por exemplo, se aplicarmos *Esquerda* para o estado inicial na figura, o estado resultante terá comutado o 5 e o branco.



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

- **Ações:** A formulação mais simples define as ações como movimentos do quadrado vazio, *Esquerda, Direita, Para Cima ou Para Baixo*.
- **Modelo de Transição:** (não pode ser esquecido enquanto você estiver produzindo o programa de agente) Dado um estado e ação, ele devolve o estado resultante; por exemplo, se aplicarmos *Esquerda* para o estado inicial na figura, o estado resultante terá comutado o 5 e o branco.
- **Teste de objetivo:** Verifica se o estado corresponde à configuração de estado objetivo mostrada na figura.



Resolução de Problemas por meio de Busca

Breve exemplificação com outro problema:

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

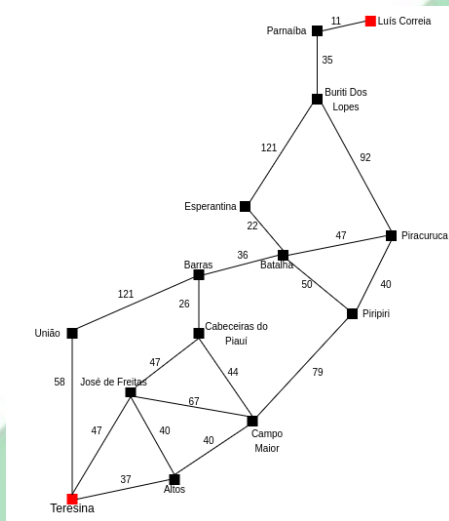
Estado objetivo

- **Modelo de Transição:** (não pode ser esquecido enquanto você estiver produzindo o programa de agente) Dado um estado e ação, ele devolve o estado resultante; por exemplo, se aplicarmos *Esquerda* para o estado inicial na figura, o estado resultante terá comutado o 5 e o branco.
- **Teste de objetivo:** Verifica se o estado corresponde à configuração de estado objetivo mostrada na figura.
- **Custo de caminho:** Cada passo custa 1 e, assim, o custo do caminho é o número de passos do caminho.



Estratégias de Busca

■ Voltemos ao nosso problema original!



Estratégias de Busca

- Como se trata de um grafo, podemos utilizar uma busca em árvore (atenção que eu vou desenhar no quadro).



Estratégias de Busca

- Como se trata de um grafo, podemos utilizar uma busca em árvore (atenção que eu vou desenhar no quadro).
- Como fazer para não repetir um estado?



Estratégias de Busca

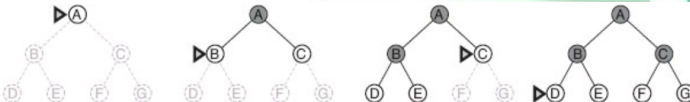
- Como se trata de um grafo, podemos utilizar uma busca em árvore (atenção que eu vou desenhar no quadro).
- Como fazer para não repetir um estado?
- Como construir o caminho?



Estratégias de Busca

- Como se trata de um grafo, podemos utilizar uma busca em árvore (atenção que eu vou desenhar no quadro).
- Como fazer para não repetir um estado?
- Como construir o caminho?

■ Busca em Largura



- **Busca de custo uniforme:** em vez de expandir o nó mais raso, é expandido o nó com o menor custo de caminho.
- **Busca em Profundidade,** a qual pode ser modificada para busca com profundidade limitada e aprofundamente iterativo.



Busca com Informação



Estácio

Busca com Informação (Heurística)

- Com a adição de informação, podemos melhorar nossa estratégia de busca, através da adição de uma **função de avaliação** — ou seja, uma forma de medir o quão boa é uma determinada solução.



Busca com Informação (Heurística)

- Com a adição de informação, podemos melhorar nossa estratégia de busca, através da adição de uma **função de avaliação** — ou seja, uma forma de medir o quão boa é uma determinada solução.
- Utilizando o nosso exemplo, vamos levar em consideração que agora sabemos a distância **em linha reta** entre cada cidade e o destino que queremos alcançar. Então teremos a seguinte tabela:

Teresina	277	Esperantina	129
Altos	255	Batalha	134
União	231	Piripiri	155
José de Freitas	231	Piracuruca	117
Barras	167	Buriti dos Lopes	40
Cabeceiras do Piauí	191	Parnaíba	10
Campo Maior	223	Luís Correia	0

Tabela: Distância em linha reta para Luís Correia



Busca Gulosa, ou *Greed Search*

- O algoritmo **Busca Gulosa** procura pelo “vizinho mais próximo”.

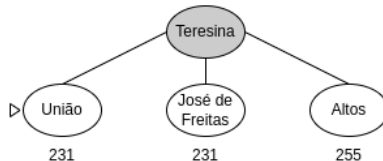


Busca Gulosa, ou *Greed Search*

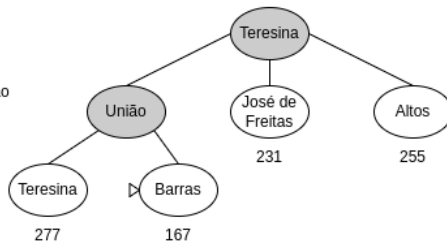
Estado Inicial



Após expandir Teresina



Após expandir União



Busca Gulosa, ou *Greed Search*

- A continuação seria:
 - Barras → Batalha → Piracuruca → Buriti dos Lopes → Parnaíba → Luís Correia.
- A distância total (a partir de Teresina): $58 + 121 + 36 + 47 + 92 + 35 + 11 = 400\text{km}$.



Busca Gulosa, ou *Greed Search*

- A continuação seria:
 - Barras → Batalha → Piracuruca → Buriti dos Lopes → Parnaíba → Luís Correia.
- A distância total (a partir de Teresina): $58 + 121 + 36 + 47 + 92 + 35 + 11 = 400\text{km}$.
- É o caminho mais curto?



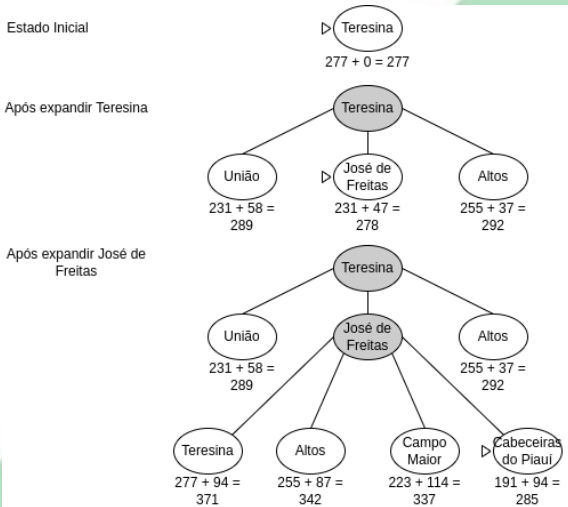
Busca Gulosa, ou *Greed Search*

- A continuação seria:
 - Barras → Batalha → Piracuruca → Buriti dos Lopes → Parnaíba → Luís Correia.
- A distância total (a partir de Teresina): $58 + 121 + 36 + 47 + 92 + 35 + 11 = 400\text{km}$.
- É o caminho mais curto?
- É possível que não! Vamos testar outro algoritmo.



A^*

- Este algoritmo (*a estrela*) utiliza, além do conhecimento que adquirimos, o custo para se alcançar cada estado.



A*

- É possível que o A* demore mais para encontrar uma solução em relação ao Algoritmo Guloso, porém é garantido que a solução será a melhor.



A*

- É possível que o A* demore mais para encontrar uma solução em relação ao Algoritmo Guloso, porém é garantido que a solução será a melhor.
- Numa situação de “mundo real”, pode acontecer do A* demorar muito, muito mesmo, devido à quantidade de estados possíveis.



A*

- É possível que o A* demore mais para encontrar uma solução em relação ao Algoritmo Guloso, porém é garantido que a solução será a melhor.
- Numa situação de “mundo real”, pode acontecer do A* demorar muito, muito mesmo, devido à quantidade de estados possíveis.
- Por causa disso existem outros algoritmos de busca, os quais são modificações ou inspirados no A*: IDA* (*Iterative Deepening A** — Aprofundamento Iterativo A*) e RBFS (*Recursive Best First Search* — Busca Recursiva de Melhor Escolha).



A*

- É possível que o A* demore mais para encontrar uma solução em relação ao Algoritmo Guloso, porém é garantido que a solução será a melhor.
- Numa situação de “mundo real”, pode acontecer do A* demorar muito, muito mesmo, devido à quantidade de estados possíveis.
- Por causa disso existem outros algoritmos de busca, os quais são modificações ou inspirados no A*: IDA* (*Iterative Deepening A** — Aprofundamento Iterativo A*) e RBFS (*Recursive Best First Search* — Busca Recursiva de Melhor Escolha).
- Tarefa de casa: pesquisar sobre esses e outros algoritmos!



Terminamos a Terceira Parte!
Obrigado pela atenção!

