

Planejamento

Agentes que exploram espaços de estados

Prof. Marcelo de Souza

85ECS – Engenharia de Software Orientada a Agentes

Universidade do Estado de Santa Catarina

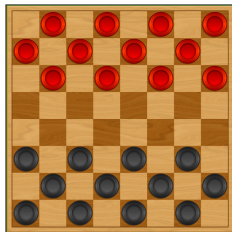
Problemas de planejamento



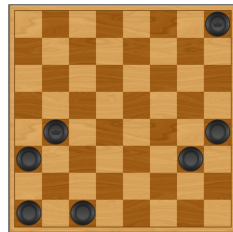
Problemas de planejamento



Agente jogador
de damas



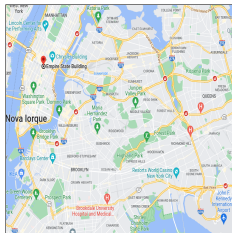
Como vencer?



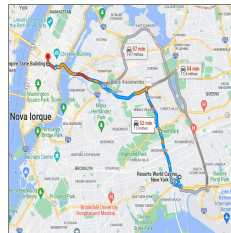
Problemas de planejamento



Agente roteador
(GPS)



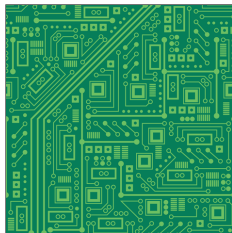
Como chegar?



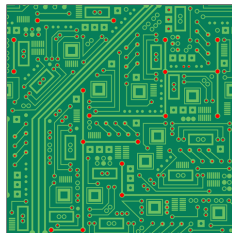
Problemas de planejamento



Agente furador
de circuitos
integrados



Como furar?





Problema de planejamento: definir o plano, i.e. a sequência de ações que o agente deve executar, para transformar o estado atual no (ou mais próximo do) estado desejado.



Problema de planejamento: definir o plano, i.e. a sequência de ações que o agente deve executar, para transformar o estado atual no (ou mais próximo do) estado desejado.

Para resolver problemas de planejamento, usamos **algoritmos de busca**. Esses algoritmos exploram o espaço de estados em busca do caminho até o estado desejado.

Busca em espaços de estados



Algoritmos de busca se baseiam em:

- ▶ Estados (ambiente);
- ▶ Ações (agente);
- ▶ Transições (efeito das ações nos estados e seus custos);
- ▶ Objetivo



Algoritmos de busca se baseiam em:

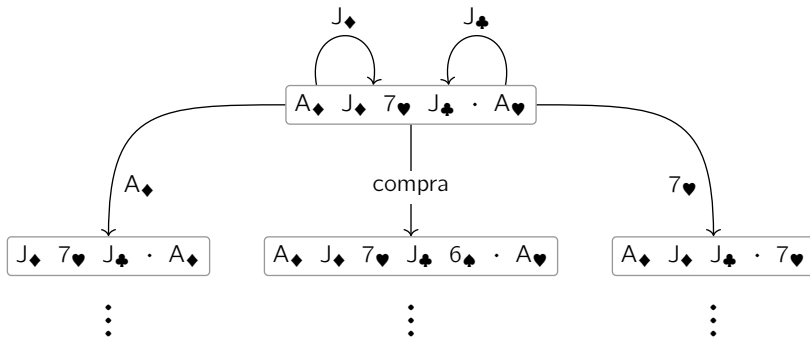
- ▶ Estados (ambiente);
- ▶ Ações (agente);
- ▶ Transições (efeito das ações nos estados e seus custos);
- ▶ Objetivo

Exemplo – jogo do descarte

- ▶ **Estados:** cartas da mão + carta aberta;
- ▶ **Ações:** descartar, comprar;
- ▶ **Transições:**
 - ▶ descartar → remove carta da mão e atualiza carta aberta;
 - ▶ comprar → remove carta do monte e insere na mão.
- ▶ **Objetivo:** esvaziar a mão.

Jogo do descarte

Estados, ações e transições





Modelagem

- ▶ **Vértices** representam os estados do problema;
- ▶ **Arcos** representam as transições entre estados.

Conceitos

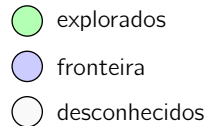
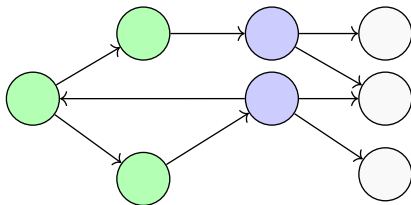
- ▶ Vértices **explorados**: estados já expandidos;
- ▶ Vértices de **fronteira**: estados candidatos à expansão;
- ▶ Vértices **desconhecidos**: estados que ainda não estão na memória.

Modelagem

- ▶ **Vértices** representam os estados do problema;
- ▶ **Arcos** representam as transições entre estados.

Conceitos

- ▶ Vértices **explorados**: estados já expandidos;
- ▶ Vértices de **fronteira**: estados candidatos à expansão;
- ▶ Vértices **desconhecidos**: estados que ainda não estão na memória.



Algoritmo básico de busca

Seja:

- ▶ X os vértices explorados; F os vértices de fronteira;
- ▶ s o vértice inicial; $N(v)$ os sucessores do vértice v .

Algoritmo básico de busca

$X \leftarrow \{\}$

$F \leftarrow \{s\}$

enquanto $F \neq \{\}$ **faça**

$v \leftarrow$ retira um vértice de F

se v é o objetivo **então**

 └ **retorna** o plano/caminho $s \rightsquigarrow v$

se $v \notin X$ **então**

 └ $X \leftarrow X \cup \{v\}$

 └ $F \leftarrow F \cup N(v)$

retorna falha

Algoritmo básico de busca



Seja:

- ▶ X os vértices explorados; F os vértices de fronteira;
- ▶ s o vértice inicial; $N(v)$ os sucessores do vértice v .

Algoritmo básico de busca

$X \leftarrow \{\}$

$F \leftarrow \{s\}$

enquanto $F \neq \{\}$ **faça**

$v \leftarrow$ retira um vértice de F

se v é o objetivo **então**

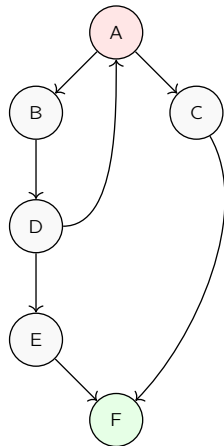
retorna o plano/caminho $s \rightsquigarrow v$

se $v \notin X$ **então**

$X \leftarrow X \cup \{v\}$

$F \leftarrow F \cup N(v)$

retorna falha



Ideia: expandir o grafo em camadas a partir do vértice inicial.

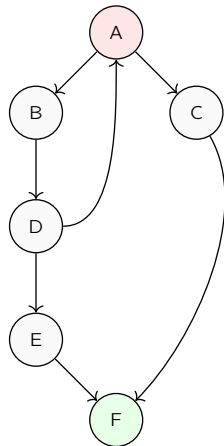
- ▶ Ou seja, avaliar vértices a um salto do vértice inicial, depois aqueles a dois saltos, etc.
- ▶ **Implementação:** fronteira F é uma fila (FIFO).

Ordem de visitação: A, B, C, D, E;

Solução: A, C, F.

Análise

- ▶ Completa;
- ▶ Ótima, se os custos são uniformes.



Ideia: expandir o grafo o mais distante possível do vértice inicial.

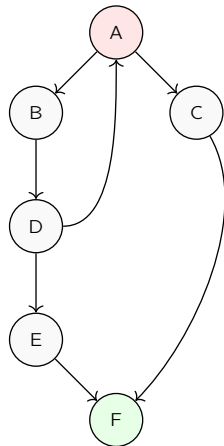
- ▶ Ou seja, avaliar primeiro vértices descobertos recentemente.
- ▶ **Implementação:** fronteira F é uma pilha (LIFO).

Ordem de visitação: A, B, D, E, F;

Solução: A, B, D, E, F.

Análise

- ▶ Completa se o grafo é finito;
- ▶ Não é ótima.



Busca de custo uniforme



Ideia: expandir o grafo se afastando da origem com custo mínimo.

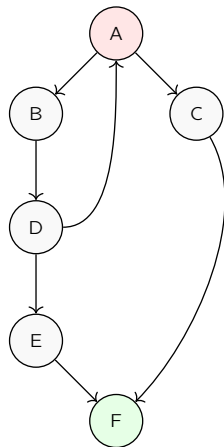
- ▶ Ou seja, avaliar primeiro vértices de menor custo.
- ▶ **Implementação:** fronteira F é uma fila de prioridades (heap).

Ordem de visitação: A, B, C, D, F;

Solução: A, C, F.

Análise

- ▶ Completa;
- ▶ Ótima, se os pesos são positivos e não há caminhos infinitos de custo zero.



Ideia: expandir o grafo se afastando da origem com custo mínimo.

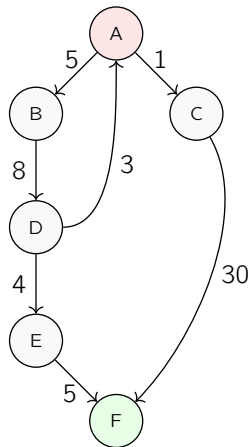
- ▶ Ou seja, avaliar primeiro vértices de menor custo.
- ▶ **Implementação:** fronteira F é uma fila de prioridades (heap).

E com custos não uniformes? A, C, B, D, E, F

Solução: A, B, D, E, F.

Análise

- ▶ Completa;
- ▶ Ótima, se os pesos são positivos e não há caminhos infinitos de custo zero.



85ECS – Engenharia de Software Orientada a Agentes
Prof. Marcelo de Souza