Inteligência Computacional

Rafael D. Ribeiro, M.Sc.
rafaeldiasribeiro@gmail.com
http://www.rafaeldiasribeiro.com.br

Inteligência Computacional

Métodos Informados de Busca

Busca Gulosa

Algoritmo A*

Algoritmo A* iterativo em profundidade

Inteligência Computacional

Algoritmo A*

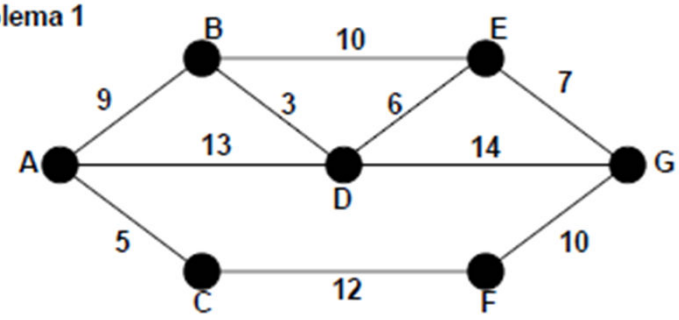
- A busca ordenada seleciona o nó com o menor custo associado da origem até ele. Já a busca gulosa escolhe o nó com a menor estimativa de custo a partir dele até uma solução.
- O algoritmo A* combina as duas estratégias visando tirar proveito das características de ambas: soluções ótimas e capacidade de explorar todo o espaço de busca (busca ordenada), e rapidez e relativamente pouca necessidade de memória (busca gulosa).
- Esta combinação ocorre através da seguinte função de avaliação:

$$f(n) = g(n) + h(n)$$

em que:
g(n) é a função que retorna o custo real da origem até o nó n
h(n) é a função heurística que retorna o custo estimado a partir do nó até a solução

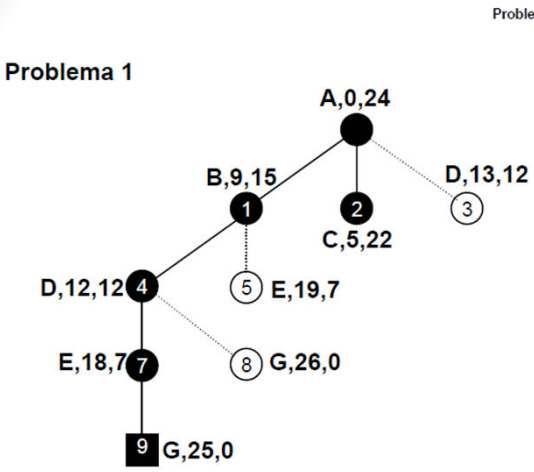
Inteligência Computacional

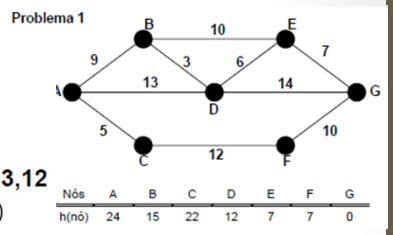
Problema 1



Nós	A	B	C	D	E	F	G
h(nó)	24	15	22	12	7	7	0

Inteligência Computacional





Inteligência Computacional

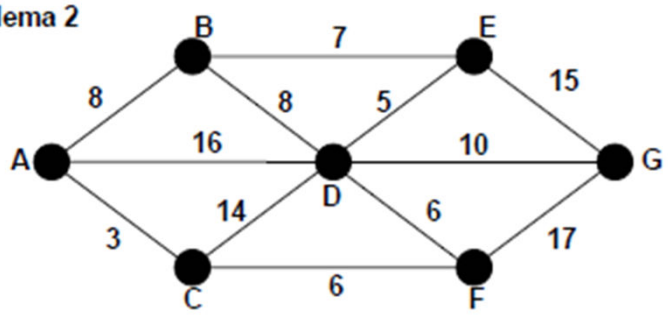
Algoritmo A*

- Os nós que foram criados e posteriormente descartados estão indicados pelos arcos tracejados. Os números ao lado das letras indicam, respectivamente, o custo real até um determinado nó e o valor da função heurística para ele.
- A numeração dentro dos nós indica a ordem de geração dos mesmos. Observa-se que o uso de informação sobre o problema permitiu que o método produzisse menos nós que a busca ordenada.

- Se as funções estiverem bem definidas, o A* é eficiente e sempre encontrará a solução ótima de um problema se este possuir alguma. Para que isso ocorra, um cuidado a ser tomado é que a função *h(n). N*ão deve superar o custo real do caminho entre um nó e a solução.

Inteligência Computacional

Problema 2

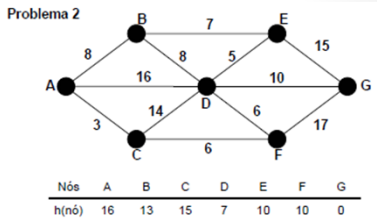


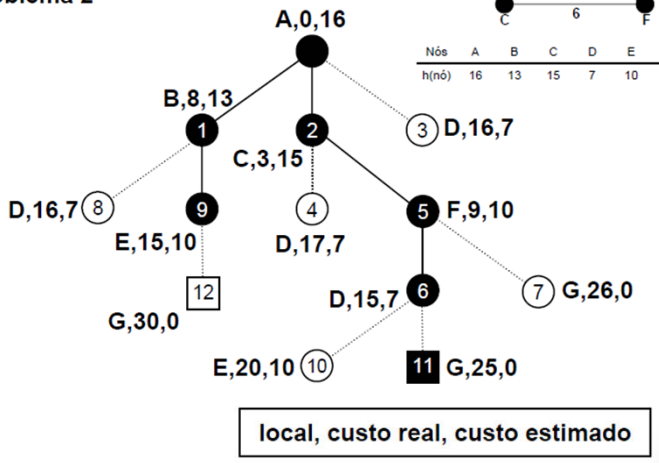
Nós	A	B	C	D	E	F	G
h(nó)	16	13	15	7	10	10	0

Inteligência Computacional

Algoritmo A*

Problema 2





local, custo real, custo estimado

Inteligência Computacional

Algoritmo A*

- Para um mesmo problema várias estimativas de custos de nós podem ser definidas, havendo diferenças em desempenho. Por exemplo, suponha que um problema possua duas heurísticas, h1 e h2, tais que h2 ≥ h1 para todo nó. Então, diz-se que a heurística h2 é **mais informada** que a h1.

- Como consequência, a árvore gerada por h2 é na média menor do que a obtida por h1. Assim, ao se utilizar de heurísticas mais informadas, o algoritmo A* obtém maior eficiência.

Inteligência Computacional

A* iterativo em profundidade

- Mesmo sendo eficiente, há muitas situações em que o algoritmo A* enfrenta dificuldades em termos de espaço disponível em memória e de tempo de processamento.

- Assim, foram desenvolvidas algumas estratégias para contornar este problema.

  - Uma das primeiras é o algoritmo A* iterativo em profundidade ou simplesmente IDA*. A cada iteração, o método aplica o A* em uma profundidade diferente até ser encontrada a solução.

Inteligência Computacional

A* iterativo em profundidade

- O conceito de profundidade aqui utilizado é totalmente diferente do que foi empregado anteriormente. A **profundidade está agora relacionada a avaliações de custos**. Pode-se pensar a **profundidade como um limite máximo para os custos**. Os nós de uma árvore cujos custos estão além desse limite são criados, avaliados e descartados.

- Na primeira iteração, a profundidade máxima ou o limite máximo de custos é a avaliação do nó raiz. Este é expandido e seus filhos avaliados.

  - Os nós que tiverem **avaliações superiores ao limite estipulado são descartados**.
  - Os outros nós são mantidos e se nenhum deles for a solução, então **aquele que tiver a melhor avaliação, ou seja, o menor custo, é expandido do mesmo modo que o nó raiz**.

Inteligência Computacional

A* iterativo em profundidade

- O processo continua expandindo os nós e mantendo apenas aqueles de avaliações não maiores que o limite estipulado até que a solução seja encontrada, quando então o algoritmo termina, ou até que toda a árvore seja gerada sem encontrá-la. Neste caso, o algoritmo realiza uma nova iteração.

- Antes de o IDA* iniciar efetivamente mais uma iteração, ele redefine o limite máximo para expansão. O novo valor passa a ser a menor das avaliações dos nós descartados na última iteração. A árvore que foi gerada anteriormente é descartada e uma nova é construída do mesmo modo utilizando o novo valor de limite máximo ou profundidade.

- Observe que o aumento da profundidade ocasiona uma árvore maior. O processo de redefinição de limite, de geração e de descarte de árvores se repete até que a solução seja encontrada. Observa-se que, quando um nó solução for utilizado para definir o próximo limite, o algoritmo para pois ele já determinou um caminho ótimo do nó raiz até a solução.

Inteligência Computacional

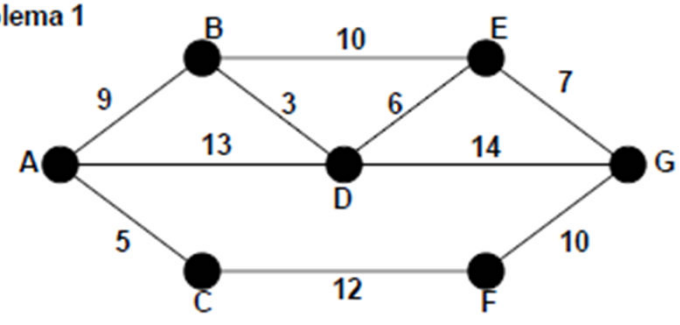
A* iterativo em profundidade

```
resposta = nulo
sucesso = falso
limite-máximo = avaliação do estado inicial
enquanto (sucesso = falso) faça
        resposta e sucesso = A* com limite-máximo
        se (sucesso = falso)
         então   limite-máximo = melhor avaliação dos nós descartados
        fim-se
fim-enquanto
retorna sucesso e reposta
```

Inteligência Computacional

A* iterativo em profundidade

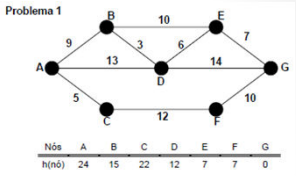
Problema 1

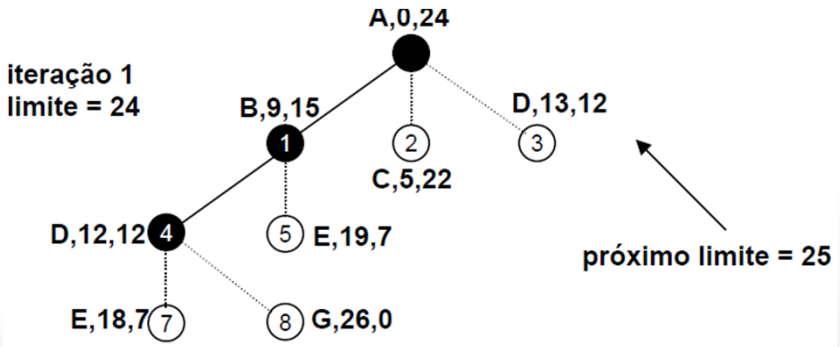


Nós	A	B	C	D	E	F	G
h(nó)	24	15	22	12	7	7	0

Inteligência Computacional

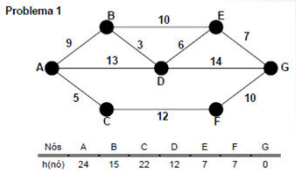
A* iterativo em profundidade

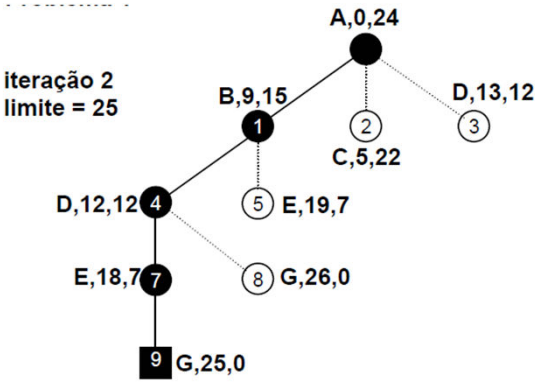




Inteligência Computacional

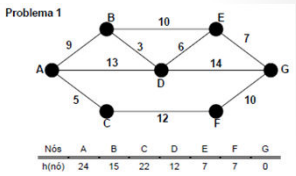
A* iterativo em profundidade

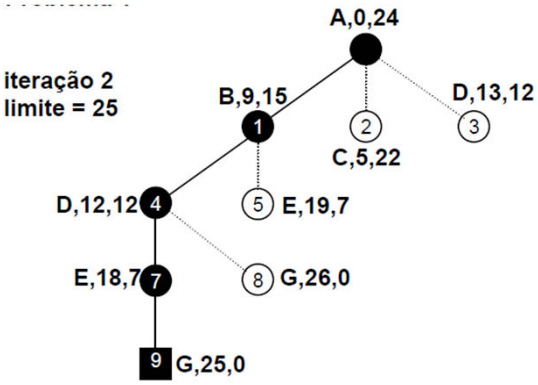




Inteligência Computacional

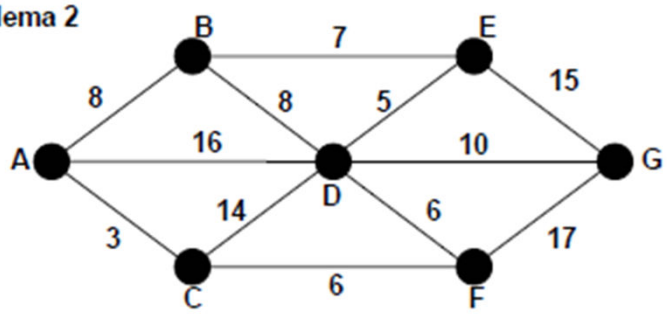
A* iterativo em profundidade





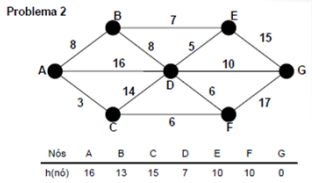
Inteligência Computacional

Problema 2

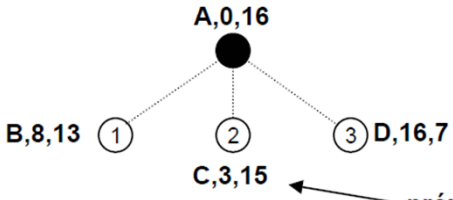


Nós	A	B	C	D	E	F	G
h(nó)	16	13	15	7	10	10	0

Inteligência Computacional

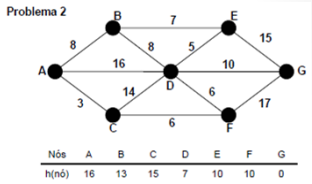


iteração 1
limite = 16

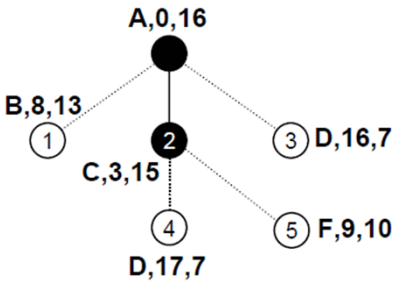


próximo limite = 18

Inteligência Computacional

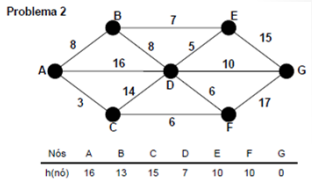


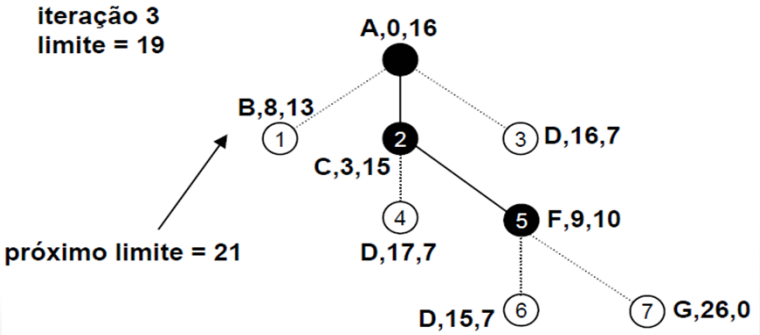
iteração 2
limite = 18



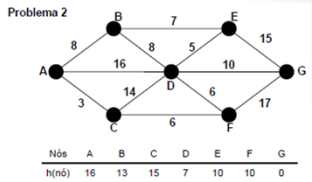
próximo limite = 19

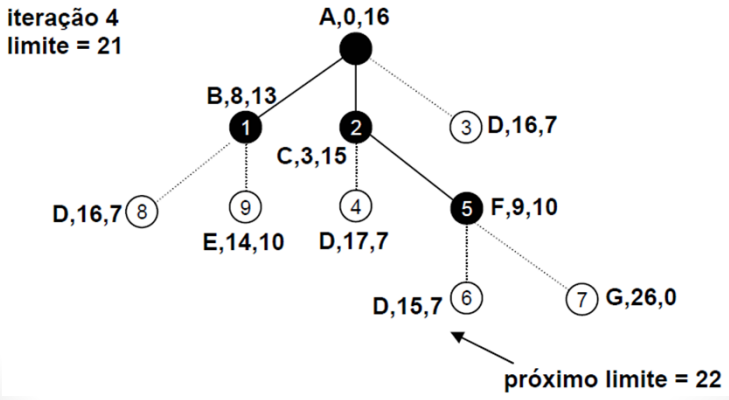
Inteligência Computacional



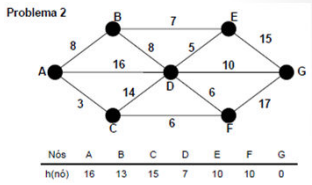


Inteligência Computacional

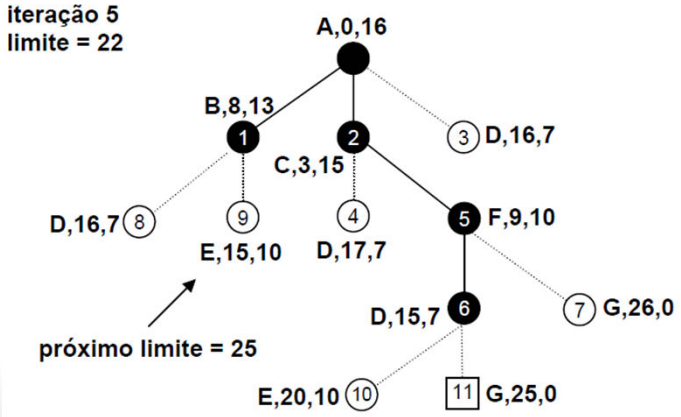




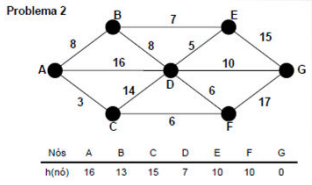
Inteligência Computacional



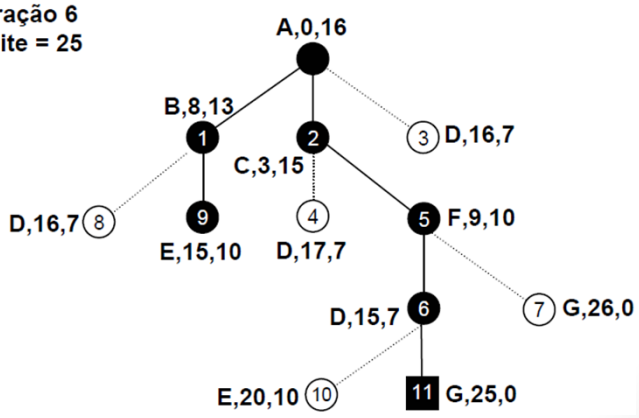
iteração 5
limite = 22



Inteligência Computacional



iteração 6
limite = 25



Inteligência Computacional

A* iterativo em profundidade

- Se as funções estiverem bem definidas, o A* é eficiente e sempre encontrará a solução ótima de um problema se este possuir alguma.
- Para que isso ocorra, um cuidado a ser tomado é que a função *h(n)* não deve superar o custo real do caminho entre um nó e a solução.

Inteligência Computacional

- Bibliografia utilizada para estas notas de aula:
  - Russel, Stuart J. Inteligência Artificial. Stuart J. Russel. Rio de Janeiro. Elsevier. 2004
  - Notas de Aula do Prof. ROGÉRIO ESPÍNDOLA, disponível na Biblioteca Virtual de docentes – SIA – Estácio, para a disciplina de Inteligência Computacional 1