




Plano de Ensino



- Apresentação da Disciplina.
- Introdução à Inteligência Artificial.
- Agentes Inteligentes.
- Resolução de Problemas.
- **Mecanismos de Busca.**
- Formas de Raciocínio Artificial.
- Representação do Conhecimento.
- Redes Semânticas.
- Aquisição de Conhecimento.
- Sistemas Especialistas.
- Sistemas Multiagentes.
- Redes Neurais.
- Mineração de Dados.



Livro-Texto



- Bibliografia Básica:
 - » RUSSELL, Stuart J.. Inteligencia Artificial. 2ª ed. Rio de Janeiro: Campus - Elsevier, 2004.
- Bibliografia Complementar:
 - » LUGER, G.F.. Inteligência Artificial : Estruturas e Estratégias para a Resolução de Problemas Complexos. 4ª ed. Porto Alegre: Artmed, 2004.

4. Mecanismos de Buscas – Busca Cega



- As estratégias de busca podem ser classificadas em:
 - » *Busca sem informação ou busca cega* → não existe nenhuma informação adicional sobre estados, além daqueles fornecidos na definição do problema. Tudo que se faz é gerar sucessores e distinguir um estado objetivo de um estado não-objetivo.
 - » *Busca com informação ou busca heurística* → são estratégias de busca que sabem se um estado não-objetivo é mais promissor que outro.
- Todas as estratégias de busca se distinguem pela *ordem* em que os nós são expandidos.
 - » Busca em extensão (*Breadth-first*);
 - » Busca de custo uniforme;
 - » Busca em profundidade (*Depth-first*);
 - » Busca em profundidade limitada;
 - » Busca de aprofundamento iterativo.

4. Mecanismos de Buscas – Busca Cega



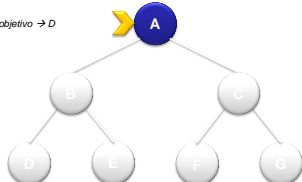
- Busca em Extensão →
 - » É uma estratégia simples em que o nó raiz é expandido primeiro, em seguida todos os sucessores do nó raiz são expandidos, depois os sucessores desses nós e assim por diante.
 - » Todos os nós em uma dada profundidade são expandidos, antes que todos os nós no nível seguinte sejam expandidos.
- Implementação:
 - » A *borda* é uma fila FIFO (*first-in, first-out*), isto é, novos itens entram no final.

4. Mecanismos de Buscas – Busca Cega



- Busca em Extensão →

Estado objetivo → D

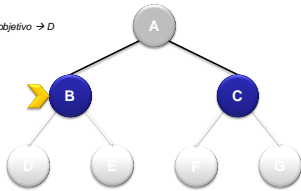


4. Mecanismos de Buscas – Busca Cega



- Busca em Extensão →

Estado objetivo → D

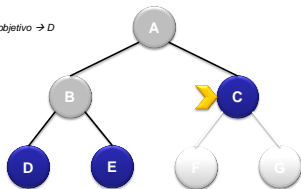


4. Mecanismos de Buscas – Busca Cega



- Busca em Extensão →

Estado objetivo → D

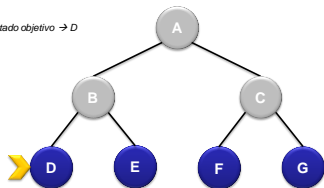


4. Mecanismos de Buscas – Busca Cega



- Busca em Extensão →

Estado objetivo → D



4. Mecanismos de Buscas – Busca Cega



- Busca em Extensão →
 - » Completa → Sim (se b é finito)
 - » Tempo → $1+b+b^2+b^3+\dots+b^d+b(b^{d-1}) = O(b^{d+1})$
 - » Espaço → $O(b^{d+1})$ (mantém todos os nós na memória)
 - » Ótima → Sim (se todas as ações tiverem o mesmo custo)
- Complexidade →

| Profundidade | Nós | Tempo | Memória |
|--------------|-----------|---------------|---------------|
| 2 | 1100 | 0,11 segundos | 1 Megabyte |
| 4 | 111.100 | 11 segundos | 106 Megabytes |
| 6 | 10^7 | 19 minutos | 10 Gigabytes |
| 8 | 10^9 | 31 horas | 1 Terabyte |
| 10 | 10^{11} | 129 dias | 101 Terabytes |
| 12 | 10^{13} | 35 anos | 10 Petabytes |
| 14 | 10^{15} | 3.523 anos | 1 Exabyte |

4. Mecanismos de Buscas – Busca Cega

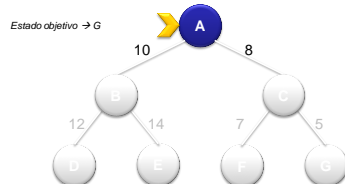


- Busca de Custo Uniforme →
 - » É uma estratégia simples em que o nó raiz é expandido primeiro, em seguida todos os sucessores com o caminho de custo mais baixo.
 - » Se todos os custos de passos forem iguais, essa busca será idêntica à busca em extensão.
 - » A busca de custo uniforme não se importa com o número de passos que um caminho tem, mas apenas com o seu custo total.
- Implementação:
 - » A *borda* é uma fila ordenada pelo custo do caminho.

4. Mecanismos de Buscas – Busca Cega



- Busca de Custo Uniforme →

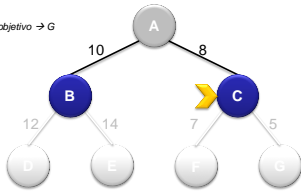


4. Mecanismos de Búsquedas – Busca Cega



- Busca de Custo Uniforme →

Estado objetivo → G

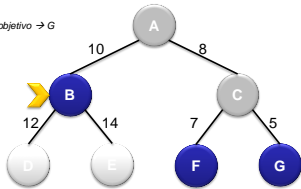


4. Mecanismos de Búsquedas – Busca Cega



- Busca de Custo Uniforme →

Estado objetivo → G

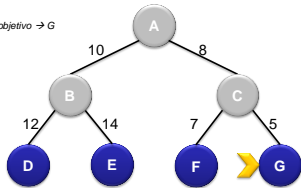


4. Mecanismos de Búsquedas – Busca Cega



- Busca de Custo Uniforme →

Estado objetivo → G



4. Mecanismos de Buscas – Busca Cega



- Busca de Custo Uniforme →
 - » Completa → Sim, se o custo de cada passo $\geq \epsilon$
 - » Tempo → # de nós com $g \leq$ custo da solução ótima, $O(b^{\lceil C/\epsilon \rceil})$ onde C é o custo da solução ótima
 - » Espaço → de nós com $g \leq$ custo da solução ótima, $O(b^{\lceil C/\epsilon \rceil})$
 - » Ótima → Sim, pois os nós são expandidos em ordem crescente de custo total.

4. Mecanismos de Buscas – Busca Cega



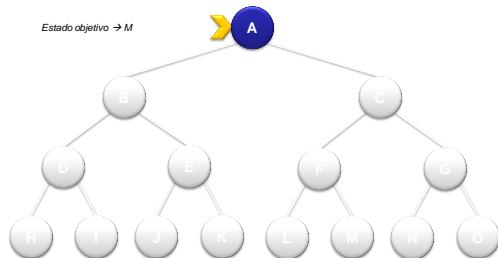
- Busca em Profundidade →
 - » É uma estratégia que expande o nó mais profundo na borda atual da árvore de busca.
 - » A busca prossegue imediatamente até o nível mais profundo da árvore de busca, onde os nós não tem sucessores.
 - » À medida que os nós são expandidos, são retirados da borda, retornando então ao nó mais raso que ainda possui sucessores inexplorados.
- Implementação:
 - » A *borda* é uma pilha LIFO (*last-in, first out*).

4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

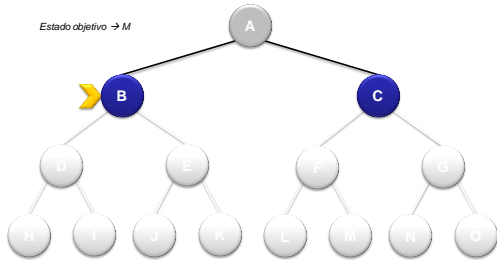


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

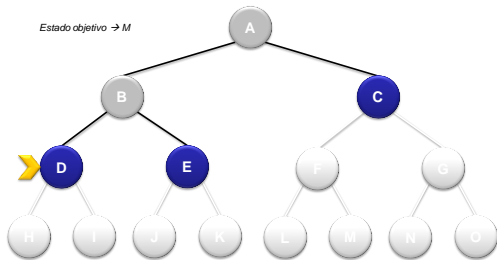


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

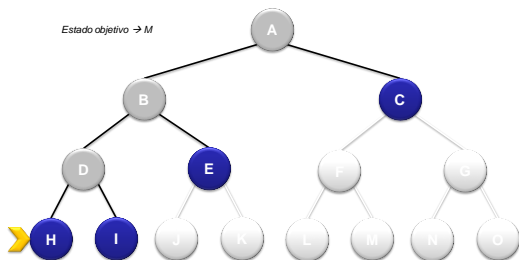


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

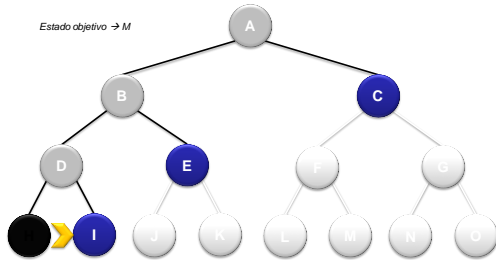


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

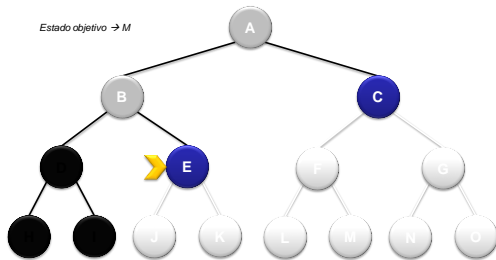


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

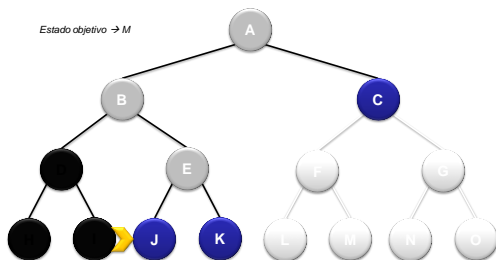


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

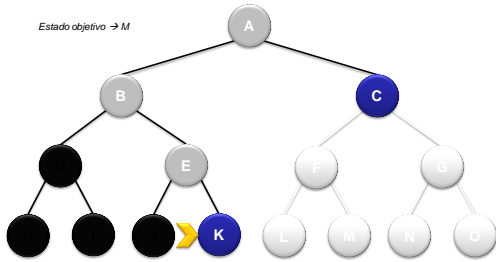


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

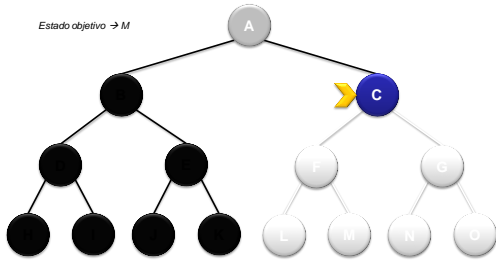


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

Estado objetivo → M

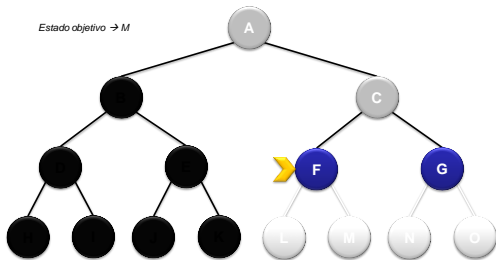


4. Mecanismos de Buscas – Busca Cega



- Busca em Profundidade →

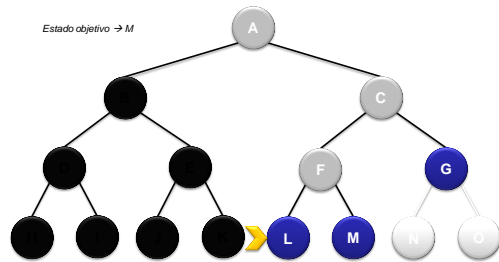
Estado objetivo → M



4. Mecanismos de Búsquedas – Busca Cega



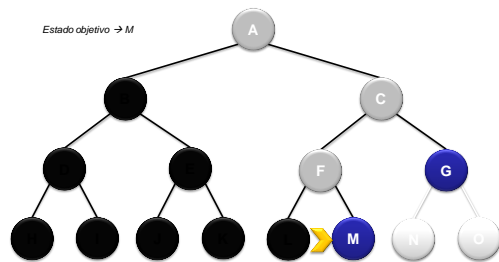
- Busca em Profundidade →



4. Mecanismos de Búsquedas – Búsqueda Ciega



- Busca em Profundidade →



4. Mecanismos de Búsquedas – Búsqueda Ciega



- Busca em Profundidade →
 - » Completa → Não: falha em espaços com profundidade infinita, espaços com loops. Se modificada para evitar estados repetidos é completa para espaços finitos.
 - » Tempo → $O(b^m)$: péssimo quando m é muito maior que d . Se há muitas soluções pode ser mais eficiente que a busca em extensão.
 - » Espaço → $O(bm)$, isto é, espaço linear. 118 KB ao invés de 10 PB para busca com profundidade $d = 12$.
 - » Ótima → Não.

4. Mecanismos de Buscas – Busca Cega

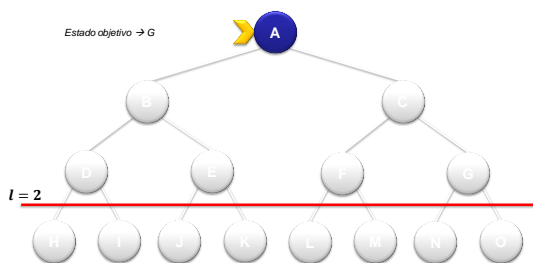


- Busca em Profundidade Limitada →
 - » Evita o problema de árvores ilimitadas, introduzindo um limite de profundidade l .
 - » Introduce uma fonte de incerteza, caso o limite de profundidade seja menor que o nível de profundidade do objetivo ($l < d$), principalmente onde d não é conhecido.
 - » Também não será ótima para os casos onde $l > d$.
 - » À medida que os nós são expandidos, são retirados da borda, retornando então ao nó mais raso que ainda possui sucessores inexplorados.
- Implementação:
 - » A borda é uma pilha LIFO (*last-in, first out*) com limite de posições.

4. Mecanismos de Buscas – Busca Cega



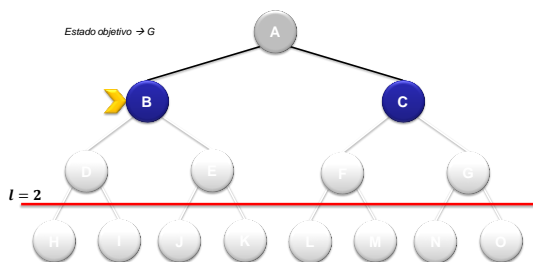
- Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



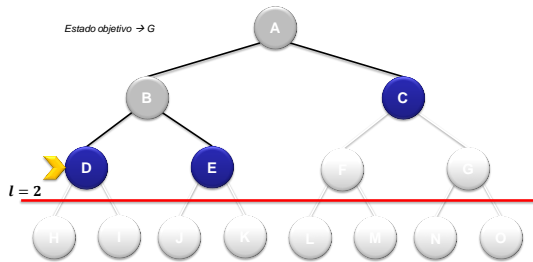
- Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



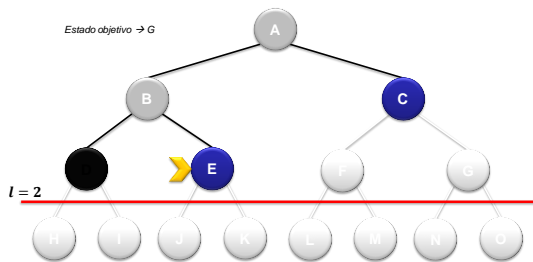
- Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



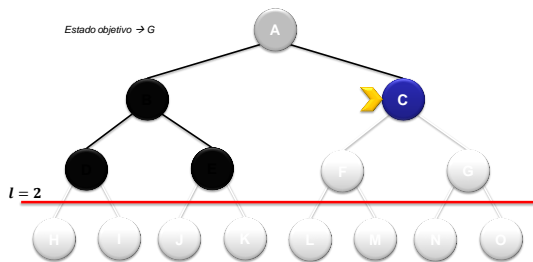
- Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



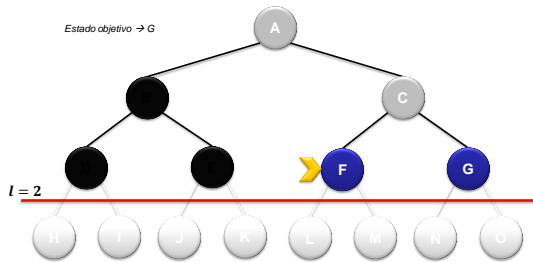
- Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



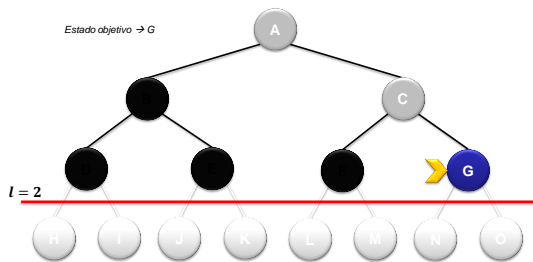
▪ Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



▪ Busca em Profundidade Limitada →



4. Mecanismos de Buscas – Busca Cega



▪ Busca em Profundidade Limitada →

- » Completa → Não; a solução pode estar além do limite.
- » Tempo → $O(b^l)$.
- » Espaço → $O(bl)$.
- » Ótima → Não.

4. Mecanismos de Buscas – Busca Cega



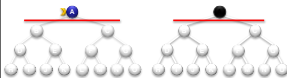
▪ Busca de Aprofundamento Iterativo →

- » É uma técnica de busca que combina a busca em profundidade com a busca por extensão.
- » A busca de aprofundamento iterativo ocupa pouca memória, fator esse que vem da busca em profundidade e é completo, pois o fator de ramificação é finito, dado pela busca por extensão.
- » O aprofundamento iterativo é o método de busca sem informação preferido quando existe um espaço de busca gerado grande e a profundidade da solução não é conhecida.
- » Apesar do *overhead* inerente, a busca de aprofundamento iterativo é eficiente já que a maior parte dos nós estará no nível inferior.

4. Mecanismos de Buscas – Busca Cega



▪ Busca de Aprofundamento Iterativo → Limite = 0

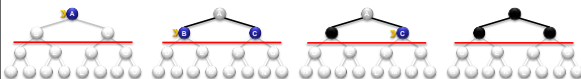


Estado objetivo → M

4. Mecanismos de Buscas – Busca Cega



▪ Busca de Aprofundamento Iterativo → Limite = 1

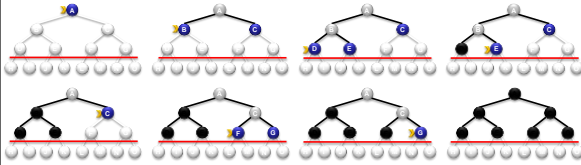


Estado objetivo → M

4. Mecanismos de Buscas – Busca Cega



- Busca de Aprofundamento Iterativo → Limite = 2

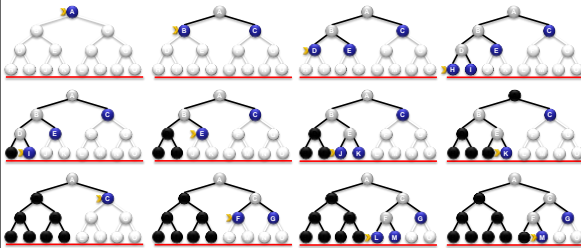


Estado objetivo → M

4. Mecanismos de Buscas – Busca Cega



- Busca de Aprofundamento Iterativo → Limite = 3



Estado objetivo → M

4. Mecanismos de Buscas – Busca Cega



- Busca de Aprofundamento Iterativo →
 - » Completa → Sim.
 - » Tempo → $(d+1)b^0 + d b^1 + (d-1)b^2 + \dots + b^d = O(b^d)$.
 - » Espaço → $O(bd)$.
 - » Ótima → Sim, se custo de passo = 1.

4. Mecanismos de Buscas – Busca Cega



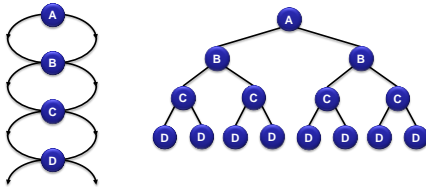
- Comparativo entre os algoritmos →

| Estratégia de Busca | Completa | Tempo | Espaço | Ótima |
|--------------------------|----------|---------------------|---------------------|-------|
| Extensão | Sim | $O(b^{d+1})$ | $O(bd+1)$ | Sim |
| Custo uniforme | Sim | $O(b^{C/\epsilon})$ | $O(b^{C/\epsilon})$ | Sim |
| Profundidade | Não | $O(b^m)$ | $O(bm)$ | Não |
| Profundidade Limitada | Não | $O(b^l)$ | $O(bl)$ | Não |
| Aprofundamento Iterativo | Sim | $O(b^d)$ | $O(bd)$ | Sim |

4. Mecanismos de Buscas – Estados Repetidos



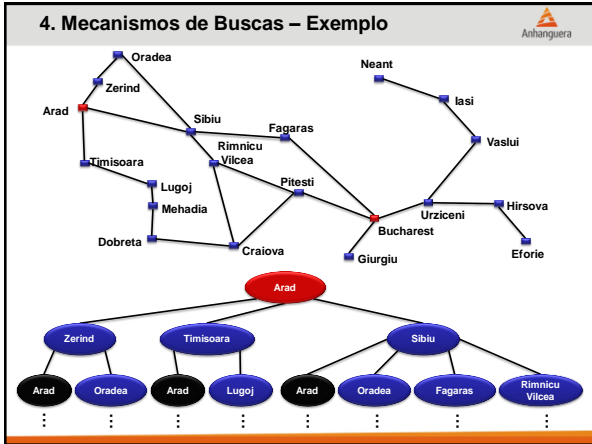
- O processo de busca pode perder tempo expandindo nós já explorados antes.
 - Estados repetidos podem levar a loops infinitos.
 - Não detectar estados repetidos pode transformar um problema linear em um problema exponencial.
 - Para alguns problemas os estados repetidos são inevitáveis: localização de rotas e quebra-cabeças, por exemplo.



4. Mecanismos de Buscas – Estados Repetidos



- Implementação → comparar os nós prestes a serem expandidos com nós já visitados.
 - Se o nó já tiver sido visitado, será descartado.
 - Lista "closed" (fechado) armazena nós já visitados.
 - Com esta lista, a busca em profundidade e a busca de aprofundamento iterativo não tem mais espaço linear.
 - A busca percorre um grafo e não uma árvore.





Inteligência Artificial

clayton.valdo@anhanguera.com