

Inteligencia Artificial

Resolución de Problemas
con
Agentes de Búsqueda

Dr. Edgar Casasola Murillo
edgar.casasola@ucr.ac.cr

Presentación en clase con fines académicos, del material de lectura de (Russell & Norvig) Artificial Intelligence A Modern Approach, Capítulos 3 y 4. (Fair use according to Copyright Act Section 107.)

Agentes de Resolución de Problemas

1. Estados del mundo atómicos (se consideran completos)
2. Se cuenta con Problemas y Soluciones bien definidas
3. Búsqueda puede ser
 - a. Des-informada
 - b. Informada
4. Solución a un problema es una secuencia fija de acciones.

Formulación del objetivo del agente

1. Se basa en:
 - a. La situación actual del agente (estado)
 - b. Medida de rendimiento
2. El objetivo es un conjunto de estados deseados del mundo.
3. El agente debe determinar los estados y acciones que considera dado el objetivo.
4. Aplica a ambientes **determinísticos** (cada acción tiene un resultado fijo)

Solución

Solución a cualquier problema **siempre** será
una **secuencia determinada** de **acciones**

Búsqueda (Search)

Proceso que explora una secuencia de acciones que alcanzar el objetivo.

Fase 1: Búsqueda - Recibe **problema** y retorna **solución**

Fase 2: Ejecución - Ejecuta la secuencia de pasos

Problemas bien definidos y soluciones

- **Estado inicial:** So
- **Acciones:** Dado un estado S retorna el conjunto de acciones que pueden ser ejecutadas en ese estado
- **Modelo de transición:** Retorna el nuevo estado en el que queda S luego de aplicar una acción particular
- **Prueba de cumplimiento del objetivo:** Verifica si estado actual cumple requisitos para ser una solución o pertenencia al conjunto de estados finales conocidos.
- **Costo de transición:** Transiciones pueden tener costos variados

Espacio de estados

Estados posibles que pueden ser generados desde el estado inicial

Grafo:

Nodos = estados

Aristas = acciones

Ruta:

Camino desde estado inicial hasta objetivo

Problemas Juguete y Vida Real

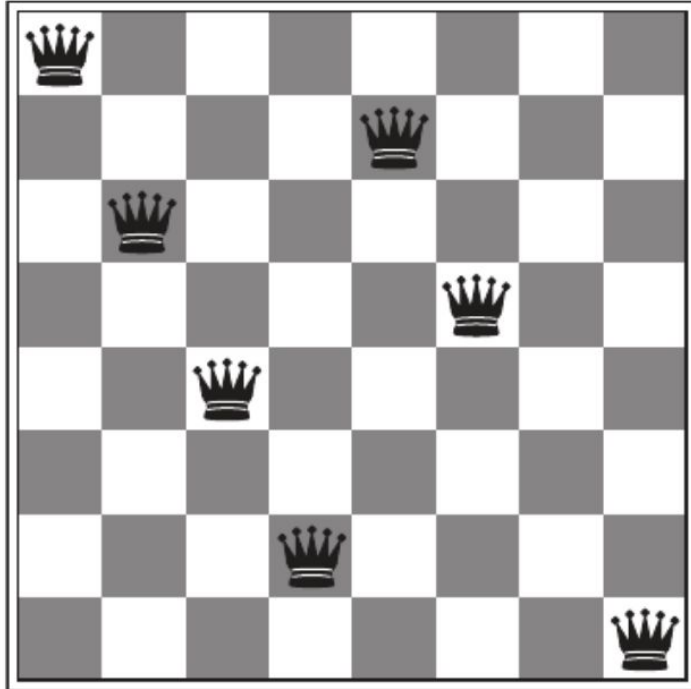
7	2	4
5		6
8	3	1

	1	2
3	4	5
6	7	8

Problemas de Bloque Corredizo (8 - Puzzle)

1. NP - Completo No hay algoritmo conocido que mejore al método de búsqueda de ruta óptima que se estudiará posteriormente
2. Usados para probar algoritmos de búsqueda.

Ocho Reinas



DOS PLANTEAMIENTOS DIFERENTES

1. **Incremental:** Estado inicial vacío
2. **Estado completo:** Ocho reinas desordenadas en posiciones aleatorias

Problema de Knuth (1964)

Encontrar la combinación de funciones de factorial, recorte hacia abajo y raíz cuadrada con la que a partir de 4 se obtiene cualquier otro número entero. Por ejemplo: 5 a partir de 4

$$\left\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right\rfloor = 5$$

Estados: números positivos

Estado inicial: 4

Acciones: Aplicar factorial, raíz cuadrada o recorte hacia abajo

Modelo de transición: Dado por la definición de cada función

Prueba de cumplimiento de objetivo: Entero positivo que se desea

IMPORTANCIA

- Espacio de estados infinito
- Similar a: generación de lenguaje, generación de expresiones matemáticas, circuitos, pruebas, programas y problemas de definición recursiva.

Problemas de la vida real

Búsqueda de ruta: Viajar de un punto A a un punto B.

Estados: Ubicación y tiempo.

El modelo de transición está determinado por el medio seleccionado para transportarse de un lugar a otro.

Acción: Tomar un medio de transporte para salir de una ciudad hacia otra.

Costo: El costo varía según el tiempo y dinero invertido para llegar de A hacia B.

Problema del turista o Touring Problem

Visitar cada ciudad al menos una vez llegando a una ciudad destino al menor costo.

Estado debe contener:

Ubicación actual + lista de ciudades visitadas, al final todas las ciudades deben estar en la lista.

Problema del Agente Viajero (variante del Touring): Cada ciudad puede ser visitada solamente una vez

Otros problemas

1. Diseño de circuitos digitales
2. Navegación de un robot
3. Secuencia de Ensamblaje de artefactos

Búsqueda - Search

Árbol de búsqueda:

- nodo raíz = estado inicial
- estados = nodos y acciones = aristas

Estrategia: seleccionar **expandir** un nodo y dejar otros para después

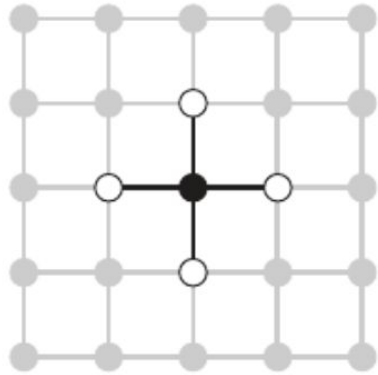
Frontera (*cc. Lista Abierta*) se almacena en una cola. Nodos a la espera de ser expandidos.

Repetición de estados => Rutas con ciclos

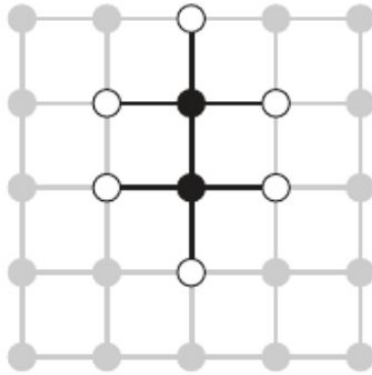
Memoria de nodos expandidos: **Conjunto de explorados** (*cc. Lista Cerrada*)

Propiedad de Separación de Regiones

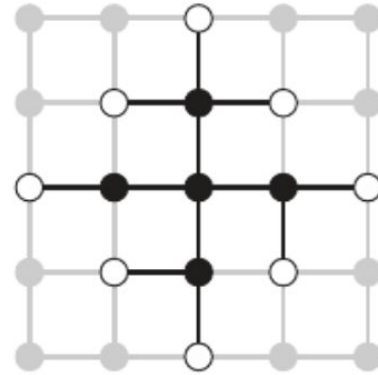
La Frontera (nodos blancos) separa la región explorada (negro) de la no explorada (gris)



(a)



(b)



(c)

Evaluación de algoritmos de búsqueda

Completitud

Garantía de encontrar solución en caso de existir

Optimalidad

La ***solución*** encontrada es ***óptima***

Complejidad temporal

Tiempo que toma encontrar la solución

Complejidad espacial

Memoria necesaria para llevar a cabo la búsqueda

Dificultad del Problema

Ciencias de la Computación

Grafo con Representación Explícita

$|\{\text{Vértices}\}| + |\{\text{Áristas}\}|$

Inteligencia Artificial

Grafo con Representación Implícita

Estado inicial + Acciones +
Transiciones

Grafo usualmente infinito

¿Cómo expresar la Complejidad?

Factor de ramificación (Branching Factor)

Cantidad Máxima **b** de Sucesores
de un nodo

Costo de búsqueda (Search Cost)

En términos de tiempo y/o memoria

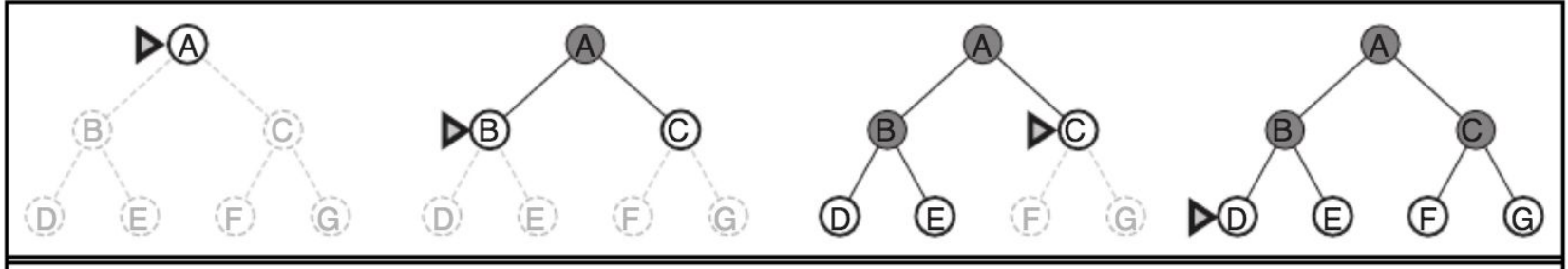
Costo total incluye costo de
búsqueda + costo de la ruta

Profundidad del Objetivo (Goal Depth)

Distancia más cercana de estado
inicial a final

Búsqueda sin información

Ancho Primero: $b=2$



$$b + b^2 + b^3 + \dots + b^d = O(b^d)$$

Requerimientos de las Heurísticas

- 1) Desigualdad triangular
- 2) Monotonicidad
- 3) Sub-estimado

Funciones Heurísticas

Costo estimado del camino más barato desde el nodo hasta la meta

- Best First
- Greedy Best - expande al más cercano a la meta
- A* utiliza una función heurística para determinar el nodo frontera que se encuentra más cerca de la solución
- ID A* o iterative deepening A* es un algoritmo profundidad primero con una profundidad tope definida por una función heurística

<https://www.git.ucr.ac.cr/ci0129/>