

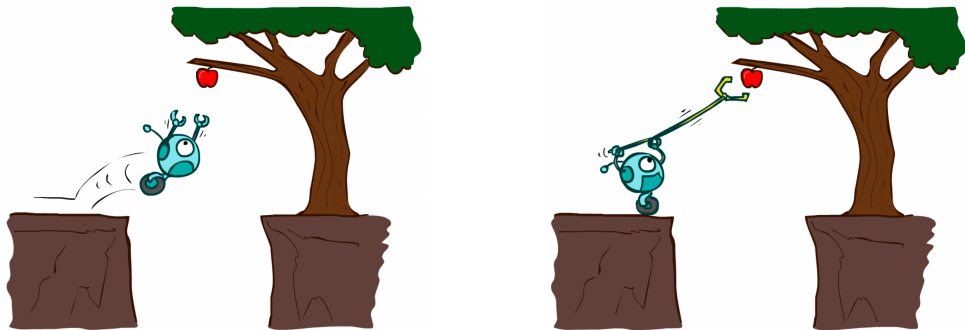
ESPACIOS DE ESTADOS

ALAN REYES-FIGUEROA
INTELIGENCIA ARTIFICIAL

(AULA 05A) 22.ENERO.2024

Problemas de Búsqueda

Cuando la acción correcta no es inmediatamente obvia: **planeamos a futuro.**



En este caso tenemos un **agente problem-solving** y el método que utiliza se llama **búsqueda**.

Problemas de Búsqueda

La búsqueda puede ser de varios tipos:

- algoritmo **no-informado** (búsqueda no informada): el agente no puede estimar qué tan lejos está del objetivo.
- algoritmo **informado** (búsqueda informada): el agente puede estimar qué tan lejos está del objetivo.
- algoritmo **priorizado** (búsqueda priorizada): existe una función de utilidad asociada decidir sobre ciertas preferencias.

Estos métodos sirven principalmente para atacar problemas de tipo

- completamente observable, determinista, estático, conocido, discreto o continuo.
- nos limitamos al caso de un solo agente.
- (En el caso desconocido, el agente no puede hacer algo mejor que actuar de forma aleatoria).

Problemas de Búsqueda

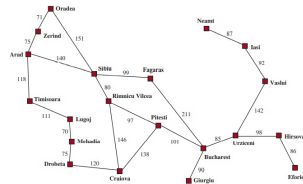
Definiciones básicas:

Requerir una solución mediante búsqueda implica:

- Formular el problema (modelo o representación)
- Definir un objetivo (inicio / fin)
- Búsqueda (simular acciones en el modelo)
- Ejecución (acciones y costos)

Ejemplo: Llegar de un lugar a otro en una red de vías.

- Representación: grafo $(G; V; E)$ de ciudades y rutas
- Inicio: S , Objetivo: G
- Búsqueda: Caminos γ a partir de S
- Costo: Distancia de γ



Problemas de Búsqueda

Problemas de búsqueda:

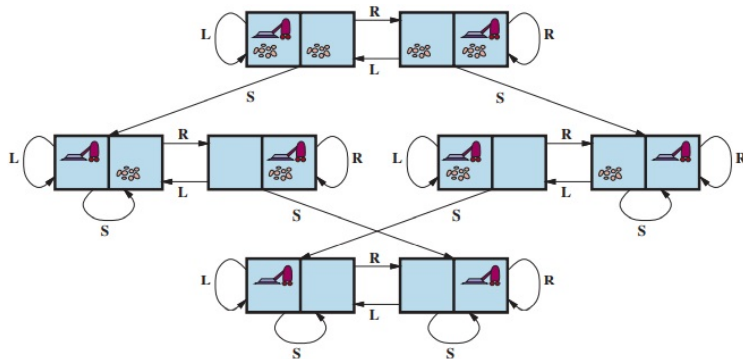
- Un espacio de estados \mathcal{S}
- Un estado inicial $S_0 \in \mathcal{S}$
- Uno o más estados objetivos ($O =$ conjunto estados objetivo)
- Acciones posibles (en cada estado)
- Modelo de transición $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. Aquí $f(S, A)$ indica el resultado de la acción a cuando estamos en el estado S .
- Función de costo $C : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. Aquí $C(S, A, S')$ indica el costo de ejecutar la acción A cuando estamos en el estado S , para llegar al estado S' .

En algunos problemas basta con hallar una solución objetivo.

En otros, buscamos soluciones óptimas: estados objetivo y que minimice la función de costo.

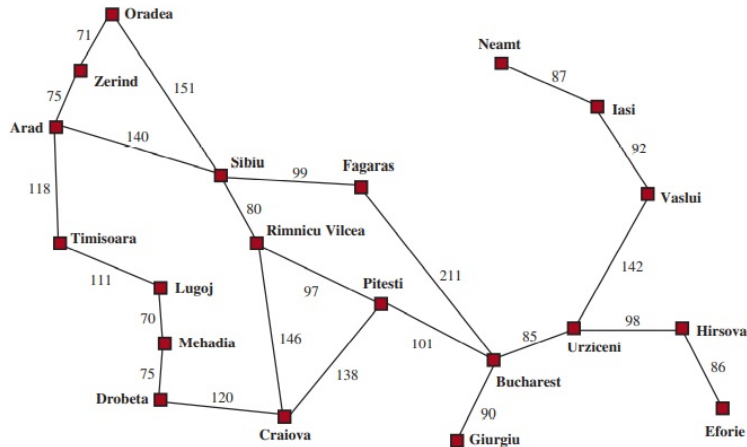
Ejemplo

Para la máquina de limpiar.



Espacio de estados de la máquina de limpiar.

Ejemplo



Problema de la ruta más corta (Shortest Path).

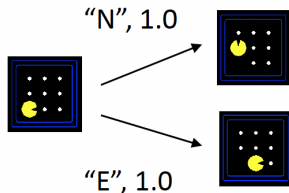
Ejemplo

Pac-man:

- Espacio de estados



- Acciones (función sucesor, costos)

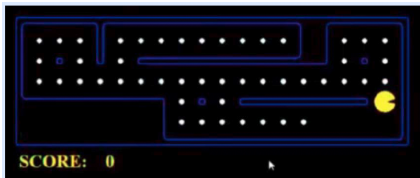


- Estados inicial y objetivo.

Una solución es una secuencia de acciones (**plan** de acción) que transforma el estado inicial en el estado objetivo.

Ejemplo

El espacio de estados almacena toda la información detallada del ambiente.



Un estado de búsqueda almacena sólo la información necesaria para el problema.

Problema de hallar una ruta (path):

- Locación (x, y)
- Acciones: NSEW
- Sucesor: sólo actualiza (x, y)
- Objetivo: ¿Es $(x, y) = END$?

Problema de comerse todos los puntos:

- $P = \{(x_i, y_i)\}$ de todos los puntos
- Acciones: NSEW
- Sucesor: actualiza (x, y) y $\{(x_i, y_i)\}$
- Objetivo: ¿Es $P = \emptyset$?

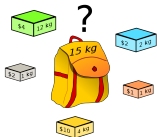
Ejemplo

8-queens:



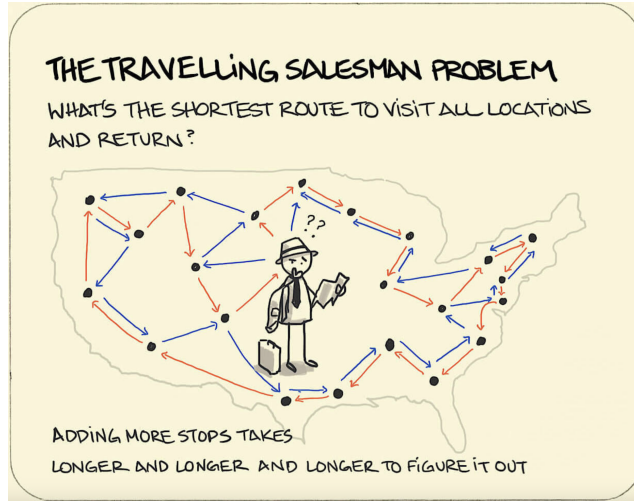
Ejemplo

Knapsack:



Ejemplo

TSP:



Ejemplo

TSP:

