Instituto Tecnológico de Buenos Aires

Sistemas de Inteligencia Artificial

Trabajo Práctico 1

Métodos de Búsqueda

Lado A - Métodos de Búsqueda

Ejercicio 1

Se cuenta con el ejercicio “8-puzzle” donde se parte de un **tablero inicial** al azar y, moviendo los numeros adyacentes al espacio vacío, se busca llegar al **tablero solución**.

Ejemplo tablero inicial:

5	7	3
8	2	
1	6	4

Tablero solución:

1	2	3
8		4
7	6	5

Para este ejercicio, no es necesaria la implementación.

Pensar:

- ¿Qué estructura de estado utilizarían?
- Al menos 2 heuristicas admisibles no-triviales
- ¿Qué métodos de búsqueda utilizarían, con qué heurística, y por qué?

Usaríamos una estructura de estado que sea una matriz con enteros, donde el espacio en blanco sea un 0. Los estados repetidos se verificarían transponiendo o rotando la matriz. Los movimientos admisibles serían poder mover el 0 hacia arriba, abajo, izquierda o derecha, restringiendo el caso que se este sobre una esquina. Por ejemplo, estando en la esquina superior derecha, solo se puede mover hacia abajo o hacía la izquierda.

Una heurística admisible sería la distancia (norma 0) de cada elemento a su correspondiente lugar. Porque siempre será necesario hacer esa cantidad de movimientos como mínimo para ubicarlos.

Otra heurística sería contar la cantidad de elementos fuera de su lugar en el tablero. Nunca sobreestimaría el costo de alcanzar la solución porque no se pueden realizar cambios entre dos casilleros llenos (se debe mover primero a uno vacío y eso hace que se requiera más de un movimiento)

Usaría un método de búsqueda informado, teniendo en cuenta estados ya explorados. De esta manera con los estados explorados puede descartar posibles estados que ya han sido evaluados y reducir el factor de ramificación. Las heurísticas que usaríamos serían las mencionadas anteriormente, porque no sobreestimarían la solución.

Ejercicio 2

Proponemos 2 juegos de los cuales deberán elegir 1 para implementar un motor de búsqueda de soluciones:

- Sokoban (https://en.wikipedia.org/wiki/Sokoban)
    - No hay restricción de cantidad de movimientos definido en el problema.
    - Queremos optimizar la cantidad de movimientos
- Grid World (multiagente)

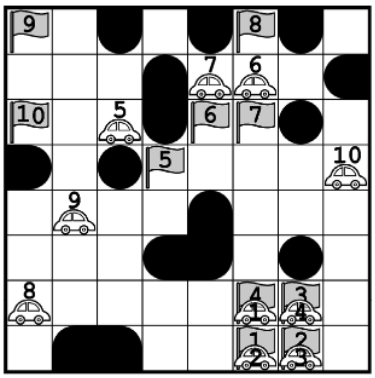

Figure 1: A small grid world instance. Cars represent initial agent positions. Flags represent destinations. Obstacles are in black.

Podemos alterar cada uno de los juegos para que sea más o menos complejo. Les aconsejamos probar con la configuración más sencilla y luego ir incrementando la complejidad.

- **Sokoban**
    - Según la configuración del tablero y la cantidad de cajas/objetivos.
- **Grid World**
    - N agentes y objetivos, y cada configuración de tablero tiene su particularidad

Implementar y resolver

- Estructura de estado
- Métodos de Búsqueda
    - BFS
    - DFS
    - Greedy
    - A*
    - IDDFS (opcional)
- Heurísticas encontradas
    - Admisibles (al menos 2)
    - No admisibles (opcional)
- Al finalizar el procesamiento…
    - Resultado (éxito/fracaso) (si es aplicable)
    - Costo de la solución
    - Cantidad de nodos expandidos
    - Cantidad de nodos frontera
    - Solución (camino desde estado inicial al final)
    - Tiempo de procesamiento
- Entregable (digital)
    - Código fuente
    - Presentación
    - Un archivo README explicando cómo ejecutar el motor de búsquedas.