

Deber-3

Los autores de nuestro texto describen uno de los métodos mas eficientes para buscar soluciones en la familia de los “n-puzzles” de un estado inicial de las piezas a un estado objetivo de las mismas. “n” en “n-puzzles” puede ser 8, 15, 24, etc. La idea principal es de generar las heurísticas mas cercanas al valor real optimo del costo del estado inicial y estados intermedios hasta el estado objetivo.

La heurística tiene que ser admisible para garantizarnos encontrar la ruta optima desde los diferentes estados inicial e intermedios hasta el estado objetivo usando el algoritmo A*. La heurística tiene que ser consistente para garantizarnos que los nodos visitados por el algoritmo A* fueron visitados por la ruta optima dado el valor de la heurística que estamos usando.

La idea principal aquí es que si tenemos un tipo de problema de búsqueda que vamos a resolver frecuentemente, vale la pena crear una base de datos con valores heurísticas cerca del valor optimo real.

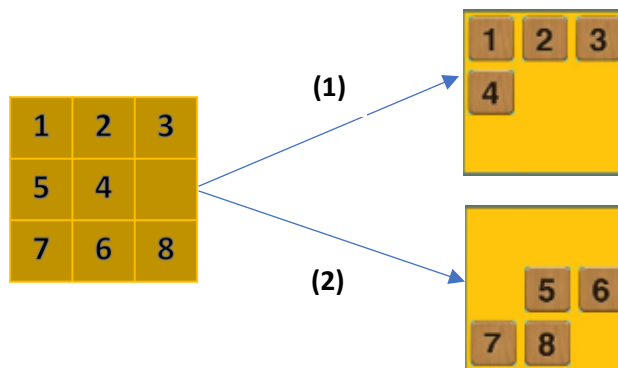
Este deber va a tratar con el “8-puzzle”. El estado objetivo será siempre el mismo:



Los estados iniciales son todos los estados que tienen solución en términos de poder llegar al estado objetivo definido arriba.

Vamos a descomponer el problema en 2 problemas más fáciles:

- (1) Si el tablero consistiera en solo cuatro cuadrillas; 1, 2, 3, y 4, con las demás cuadrillas del tablero vacías, cual fuera el número mínimo de movimientos que tomaría el ubicarles a esas cuatro cuadrillas en los lugares definidos por el estado objetivo:
- (2) Si el tablero consistiera en solo cuatro cuadrillas; 5, 6, 7, y 8, con las demás cuadrillas del tablero vacías, cual fuera el número mínimo de movimientos que tomaría el ubicarles a esas cuatro cuadrillas en los lugares definidos por el estado objetivo:



Como heurística vamos a sumar el costo de ubicar a las cuadrillas 1, 2, 3, 4 y a las cuadrillas 5, 6, 7, 8 bajo las condiciones descritas antes para cualquier combinación de cuadrillas solucionables. Esos valores serán guardados en lo que se llama una base de datos de patrones.

Usando la heurística Manhattan, que es admisible y consistente, obtendremos los costos óptimos de cada subproblema con relación al estado objetivo.

La idea es que guardaremos en un archivo los siguientes datos para cada estado del tablero solucionable:

- (a) Estado del tablero total (ubicación de todas las piezas en el tablero).
- (b) Ubicación solo de las piezas 1, 2, 3, y 4 con su costo para llegar al estado objetivo.
- (c) Ubicación solo de las piezas 5, 6, 7, y 8 con su costo para llegar al estado objetivo.

Una vez que tenemos esa “base de datos” de todos los estados solucionables del tablero podemos utilizar la suma de los costos de los subproblemas definidos en (b) y (c) para usar como heurística para cada estado que A* genera en el proceso de expansión de nodos.

Enfatizo que el costo de la ruta obtenida utilizando la heurística Manhattan y la heurística que estamos generando con el **modelo de bases de datos disjuntas** es idéntica. Lo que va a diferir grandemente va a ser el número de nodos que el algoritmo va a crear en cada instancia.

Para probar tu solución asegúrate que el costo que te da A* usando la heurística Manhattan es idéntico al costo que te da usando tu heurística en tu “base de datos”.

Una vez que tu solución esta lista prueba con diversos estados completos de tableros con estados solucionables. **Por cada estado probado:**

- (1) Usa la heurística Manhattan
 - a. Imprime el costo de la solución en términos de movimientos.
 - b. Imprime el número de nodos generados.
- (2) Usa la heurística **modelo de bases de datos disjuntas:**
 - a. Imprime el costo de la solución en términos de movimientos.
 - b. Imprime el número de nodos generados.

Entregables:

- (1) Base de datos (un archivo)
- (2) Código (incluyendo el código con las pruebas que hicieron).
- (3) Como siempre, un archivo README con el nombre de cada alumno en el grupo y lo que hizo cada uno. Describiendo también las pruebas que hicieron.

Adjunto soluciones de algunos estados solucionables del tablero para que puedan comparar:

Complete board state: (4, 5, 1, 6, 8, 7, 3, 0, 2)

DB Heuristic:

Number of nodes expanded: 330

Path cost: 21

Path: ['LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN']

Manhattan Heuristic:

Number of nodes expanded: 504

Path cost: 21

Path: ['LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN']

Complete board state: (2, 3, 5, 1, 4, 7, 0, 8, 6)

DB Heuristic:

Number of nodes expanded: 48

Path cost: 14

Path: ['RIGHT', 'UP', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN']

Manhattan Heuristic:

Number of nodes expanded: 96

Path cost: 14

Path: ['RIGHT', 'UP', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN']

Complete board state: (2, 0, 3, 1, 7, 4, 5, 8, 6)

DB Heuristic:

Number of nodes expanded: 34

Path cost: 11

Path: ['LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'RIGHT']

Manhattan Heuristic:

Number of nodes expanded: 54

Path cost: 11

Path: ['LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'RIGHT']

Complete board state: (6, 5, 1, 4, 2, 3, 0, 8, 7)

DB Heuristic:

Number of nodes expanded: 556

Path cost: 22

Path: ['UP', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT']

Manhattan Heuristic:

Number of nodes expanded: 1312

Path cost: 22

Path: ['UP', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT']

Complete board state: (3, 1, 0, 8, 4, 2, 5, 6, 7)

DB Heuristic:

Number of nodes expanded: 1542

Path cost: 24

Path: ['LEFT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN']

Manhattan Heuristic:

Number of nodes expanded: 4235

Path cost: 24

Path: ['LEFT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN']