

Práctica 2: Búsquedas informadas

Autor: Juan Rodríguez Suárez, alu0101477596@ull.edu.es.

Curso: Inteligencia Artificial, 3º Grado en Ingeniería Informática.

Fecha: 18 de octubre de 2023.

Índice

1.	Descripción del algoritmo A* implementado.....	3
2.	Tablas resultado	4
2.1.	Distancia de Manhattan	4
2.2.	Distancia euclidiana.....	8
2.3.	Conclusión	12

1. Descripción del algoritmo A* implementado

Al inicio del algoritmo, se instancia una estructura de datos tipo Arbol que va a almacenar el Nodo *head* del Arbol así como una lista de los nodos generados y analizados. Cada Nodo guarda su Nodo padre, una tupla con sus coordenadas del laberinto, el coste acumulado para llegar al Nodo y su coste heurístico. A este Arbol inicial le pasamos las coordenadas de la entrada y el coste heurístico desde esas coordenadas hasta el final, generando el Nodo *head*. Posteriormente se crean dos diccionarios (*hash*): el de los nodos abiertos y el de los nodos cerrados. La clave de los diccionarios es la coordenada del Nodo y el valor el Nodo que “posee” esas coordenadas.

Para empezar las iteraciones, añadimos al diccionario de nodos abiertos la relación entre las coordenadas de la entrada y el nodo *head* del Arbol. El algoritmo no acabará hasta que o bien ya no queden nodos abiertos o bien el nodo que se está analizando es el final.

En cada iteración, se busca el Nodo de los nodos abiertos con menor coste total (acumulado más heurístico), lo eliminamos del diccionario de nodos abiertos y lo añadimos a los nodos cerrados y analizados. Ahora se realiza una comprobación para ver si las coordenadas del nodo actual coinciden con la salida; si es así, retornamos *True* y marcamos como nodo final del Arbol al nodo que se estaba analizando. Si no es así, comprobamos todas las coordenadas válidas adyacentes y realizamos una comparación por cada coordenada adyacente: si esa coordenada ya se correspondía a un nodo de los nodos abiertos, entonces se comprueba si el nuevo coste acumulado es menor que el que tenía y, en tal caso, actualizamos su coste acumulado y su Nodo padre. Si la coordenada estaba en algún Nodo de los nodos cerrados, pasamos a la siguiente iteración. Si no estaba en los nodos abiertos ni en los cerrados entonces generamos un nuevo Nodo y lo añadimos al a lista de nodos abiertos y repetimos este proceso.

Nota: Se ha priorizado el uso de diccionarios o *hashes* en lugar de listas para almacenar los nodos abiertos y cerrados porque en la práctica resultó ser mucho más eficiente para laberintos grandes sobre todo en las operaciones de búsqueda de nodos en el diccionario.

2. Tablas resultado

2.1. Distancia de Manhattan

M_1 ID 1 Filas: 11 Columnas: 10 Entrada: (5, 1) Salida: (6, 10) Camino: [(5, 1), (5, 2), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10)]

Coste: 30

Generados: [(5, 1), (4, 2), (5, 2), (6, 2), (4, 3), (5, 3), (6, 3), (7, 2), (7, 3), (6, 4), (7, 4), (6, 5), (7, 5), (6, 6), (7, 6), (6, 7), (7, 7), (5, 8), (6, 8), (7, 8), (5, 9), (6, 9), (7, 9), (6, 10)]

Analizados: [(5, 1), (5, 2), (6, 2), (5, 3), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10)]

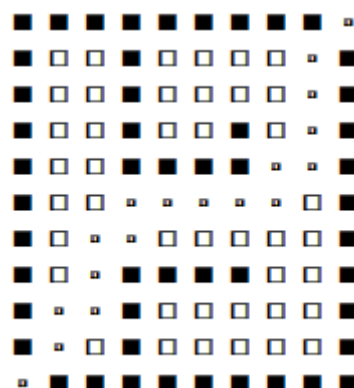


M_1 ID 2 Filas: 11 Columnas: 10 Entrada: (11, 1) Salida: (1, 10) Camino: [(11, 1), (10, 2), (9, 2), (9, 3), (8, 3), (7, 3), (7, 4), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (5, 8), (5, 9), (4, 9), (3, 9), (2, 9), (1, 10)]

Coste: 59

Generados: [(11, 1), (10, 2), (9, 2), (9, 3), (10, 3), (8, 2), (8, 3), (7, 2), (7, 3), (7, 4), (6, 2), (6, 3), (6, 4), (6, 5), (7, 5), (5, 2), (5, 3), (6, 6), (7, 6), (4, 2), (4, 3), (6, 7), (7, 7), (3, 2), (3, 3), (5, 8), (6, 8), (7, 8), (8, 8), (2, 2), (2, 3), (5, 9), (6, 9), (7, 9), (4, 8), (4, 9), (8, 9), (3, 7), (3, 8), (3, 9), (2, 7), (2, 8), (2, 9), (1, 10)]

Analizados: [(11, 1), (10, 2), (9, 2), (9, 3), (10, 3), (8, 2), (8, 3), (7, 2), (7, 3), (7, 4), (6, 2), (6, 3), (6, 4), (6, 5), (7, 5), (5, 2), (5, 3), (6, 6), (7, 6), (4, 2), (4, 3), (6, 7), (7, 7), (3, 2), (3, 3), (6, 8), (5, 8), (7, 8), (2, 2), (2, 3), (5, 9), (6, 9), (7, 9), (4, 8), (4, 9), (3, 8), (3, 9), (2, 8), (2, 9), (1, 10)]

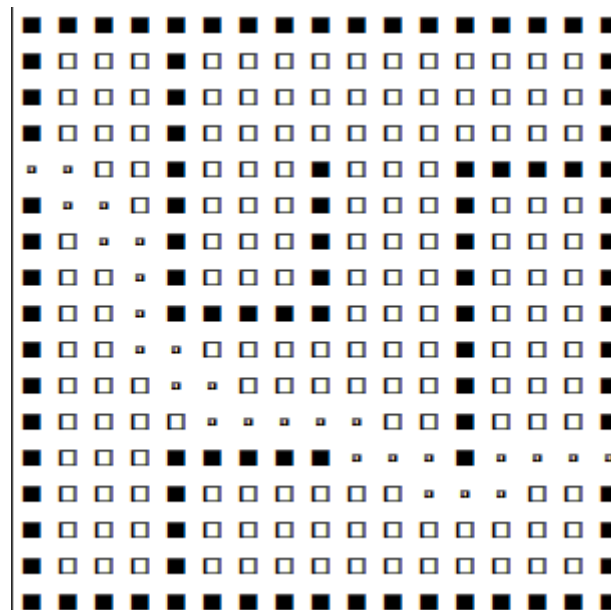


M_2 ID 1 Filas: 17 Columnas: 17 Entrada: (5, 1) Salida: (13, 17) Camino: [(5, 1), (5, 2), (6, 2), (6, 3), (7, 3), (7, 4), (8, 4), (9, 4), (10, 4), (10, 5), (11, 5), (11, 6), (12, 6), (12, 7), (12, 8), (12, 9), (12, 10), (13, 10), (13, 11), (13, 12), (14, 12), (14, 13), (14, 14), (13, 14), (13, 15), (13, 16), (13, 17)]

Coste: 78

Generados: [(5, 1), (4, 2), (5, 2), (6, 2), (4, 3), (5, 3), (6, 3), (7, 2), (7, 3), (4, 4), (5, 4), (6, 4), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (11, 2), (11, 3), (11, 4), (11, 5), (10, 6), (11, 6), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (10, 7), (11, 7), (12, 7), (13, 2), (13, 3), (13, 4), (10, 8), (11, 8), (12, 8), (14, 2), (14, 3), (14, 4), (10, 9), (11, 9), (12, 9), (9, 10), (10, 10), (11, 10), (12, 10), (13, 10), (9, 11), (10, 11), (11, 11), (12, 11), (13, 11), (14, 9), (14, 10), (14, 11), (9, 12), (10, 12), (11, 12), (12, 12), (13, 12), (14, 12), (14, 13), (3, 2), (3, 3), (3, 4), (15, 2), (15, 3), (15, 4), (8, 10), (8, 11), (8, 12), (15, 9), (15, 10), (15, 11), (15, 12), (15, 13), (13, 14), (14, 14), (15, 14), (13, 15), (14, 15), (15, 15), (12, 14), (12, 15), (12, 16), (13, 16), (14, 16), (15, 16), (13, 17)]

Analizados: [(5, 1), (5, 2), (6, 2), (5, 3), (6, 3), (7, 2), (7, 3), (5, 4), (6, 4), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (11, 2), (11, 3), (11, 4), (11, 5), (10, 6), (11, 6), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (10, 7), (11, 7), (12, 7), (13, 2), (13, 3), (13, 4), (10, 8), (11, 8), (12, 8), (10, 9), (11, 9), (12, 9), (10, 10), (11, 10), (12, 10), (13, 10), (10, 11), (11, 11), (12, 11), (13, 11), (10, 12), (11, 12), (12, 12), (13, 12), (4, 2), (4, 3), (4, 4), (14, 2), (14, 3), (14, 4), (9, 10), (9, 11), (14, 10), (14, 11), (9, 12), (14, 12), (14, 13), (14, 14), (13, 14), (13, 15), (14, 15), (13, 16), (14, 16), (13, 17)]

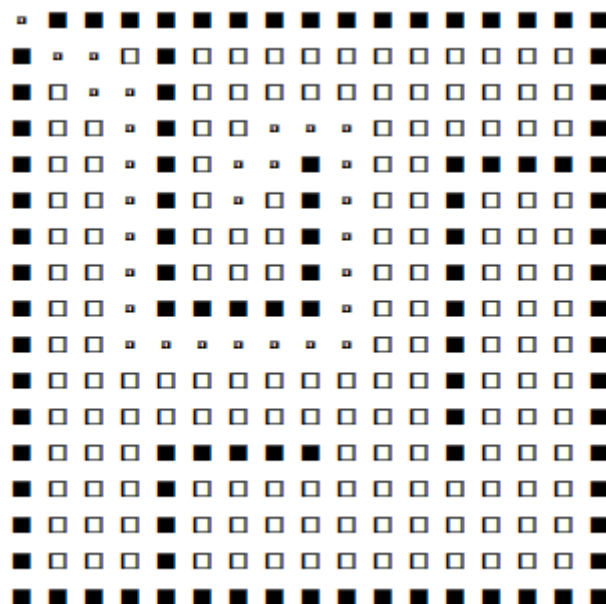


M_2 ID 2 Filas: 17 Columnas: 17 Entrada: (1, 1) Salida: (6, 7) Camino: [(1, 1), (2, 2), (2, 3), (3, 3), (3, 4), (4, 4), (5, 4), (6, 4), (7, 4), (8, 4), (9, 4), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (9, 10), (8, 10), (7, 10), (6, 10), (5, 10), (4, 10), (4, 9), (4, 8), (5, 8), (5, 7), (6, 7)]

Coste: 88

Generados: [(1, 1), (2, 2), (2, 3), (3, 2), (3, 3), (2, 4), (3, 4), (4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (11, 2), (11, 3), (11, 4), (11, 5), (10, 6), (11, 6), (10, 7), (11, 7), (10, 8), (11, 8), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (10, 9), (11, 9), (12, 9), (13, 2), (13, 3), (13, 4), (9, 10), (10, 10), (11, 10), (12, 10), (14, 2), (14, 3), (14, 4), (9, 11), (10, 11), (11, 11), (8, 10), (8, 11), (7, 10), (7, 11), (6, 10), (6, 11), (5, 10), (5, 11), (13, 10), (12, 11), (15, 2), (15, 3), (15, 4), (8, 12), (9, 12), (10, 12), (11, 12), (7, 12), (6, 12), (5, 12), (4, 9), (4, 10), (4, 11), (13, 11), (12, 12), (4, 12), (16, 2), (16, 3), (16, 4), (3, 9), (3, 10), (3, 11), (3, 8), (4, 8), (5, 8), (3, 7), (4, 7), (5, 7), (6, 7), (6, 8), (3, 6), (4, 6), (5, 6), (6, 6)]

Analizados: [(1, 1), (2, 2), (2, 3), (3, 2), (3, 3), (2, 4), (3, 4), (4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (10, 6), (10, 7), (11, 2), (11, 3), (11, 4), (11, 5), (11, 6), (11, 7), (10, 8), (11, 8), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (10, 9), (12, 8), (11, 9), (13, 2), (13, 3), (13, 4), (10, 10), (9, 10), (8, 10), (7, 10), (6, 10), (12, 9), (11, 10), (14, 2), (14, 3), (14, 4), (9, 11), (10, 11), (8, 11), (7, 11), (6, 11), (5, 10), (12, 10), (11, 11), (5, 11), (15, 2), (15, 3), (15, 4), (8, 12), (9, 12), (10, 12), (7, 12), (6, 12), (4, 10), (4, 9), (4, 8), (5, 8), (4, 7), (5, 7), (6, 7)]



M_3 ID 1 Filas: 5 Columnas: 5 Entrada: (5, 5) Salida: (2, 1) Camino: [(5, 5), (4, 5), (3, 5), (2, 5), (2, 4), (1, 4), (1, 3), (1, 2), (2, 2), (2, 1)]

Coste: 27

Generados: [(5, 5), (4, 4), (4, 5), (5, 4), (3, 5), (4, 3), (2, 4), (2, 5), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (2, 1), (3, 1)]

Analizados: [(5, 5), (4, 5), (4, 4), (5, 4), (3, 5), (4, 3), (2, 5), (2, 4), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (2, 1)]



M_3 ID 2 Filas: 5 Columnas: 5 Entrada: (5, 3) Salida: (5, 1) Camino: [(5, 3), (4, 3), (4, 4), (4, 5), (3, 5), (2, 5), (2, 4), (1, 4), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]

Coste: 43

Generados: [(5, 3), (4, 3), (4, 4), (5, 4), (4, 5), (3, 5), (2, 4), (2, 5), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]

Analizados: [(5, 3), (4, 3), (5, 4), (4, 4), (4, 5), (3, 5), (2, 5), (2, 4), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]



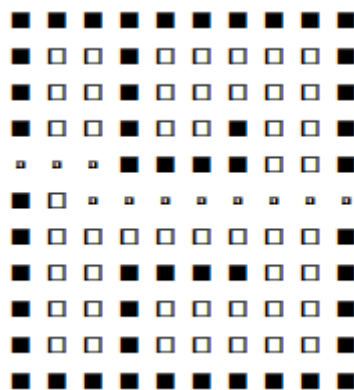
2.2. Distancia euclidiana

M_1 ID 1 Filas: 11 Columnas: 10 Entrada: (5, 1) Salida: (6, 10) Camino: [(5, 1), (5, 2), (5, 3), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10)]

Coste: 30

Generados: [(5, 1), (4, 2), (5, 2), (6, 2), (4, 3), (5, 3), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (6, 5), (7, 5), (6, 6), (7, 6), (6, 7), (7, 7), (5, 8), (6, 8), (7, 8), (5, 9), (6, 9), (7, 9), (6, 10)]

Analizados: [(5, 1), (5, 2), (5, 3), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10)]

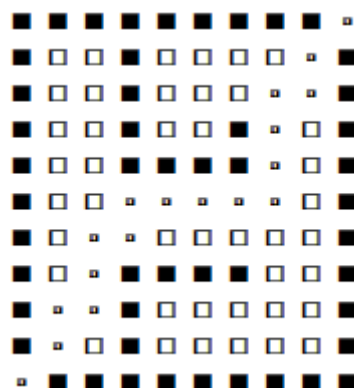


M_1 ID 2 Filas: 11 Columnas: 10 Entrada: (11, 1) Salida: (1, 10) Camino: [(11, 1), (10, 2), (9, 2), (9, 3), (8, 3), (7, 3), (7, 4), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (5, 8), (4, 8), (3, 8), (3, 9), (2, 9), (1, 10)]

Coste: 59

Generados: [(11, 1), (10, 2), (9, 2), (9, 3), (10, 3), (8, 2), (8, 3), (7, 2), (7, 3), (7, 4), (6, 2), (6, 3), (6, 4), (5, 2), (5, 3), (6, 5), (7, 5), (6, 6), (7, 6), (4, 2), (4, 3), (6, 7), (7, 7), (3, 2), (3, 3), (6, 8), (7, 8), (8, 8), (5, 8), (2, 2), (2, 3), (6, 9), (7, 9), (8, 9), (5, 9), (4, 8), (4, 9), (3, 7), (3, 8), (3, 9), (2, 7), (2, 8), (2, 9), (1, 10), (9, 7), (9, 8), (9, 9)]

Analizados: [(11, 1), (10, 2), (9, 2), (10, 3), (9, 3), (8, 2), (8, 3), (7, 2), (7, 3), (6, 2), (7, 4), (6, 3), (6, 4), (7, 5), (5, 2), (5, 3), (6, 5), (7, 6), (6, 6), (4, 2), (4, 3), (7, 7), (6, 7), (3, 2), (3, 3), (7, 8), (6, 8), (5, 8), (4, 8), (3, 8), (2, 2), (2, 3), (7, 9), (6, 9), (5, 9), (4, 9), (3, 9), (2, 8), (2, 9), (8, 8), (1, 10)]

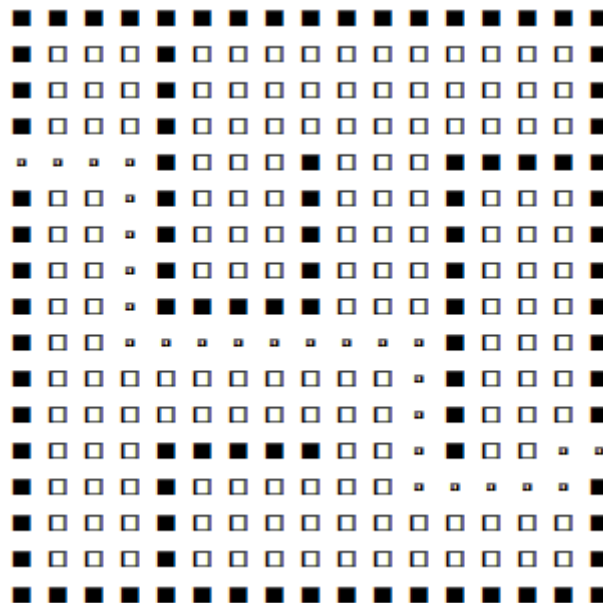


M_2 ID 1 Filas: 17 Columnas: 17 Entrada: (5, 1) Salida: (13, 17) Camino: [(5, 1), (5, 2), (5, 3), (5, 4), (6, 4), (7, 4), (8, 4), (9, 4), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (11, 12), (12, 12), (13, 12), (14, 12), (14, 13), (14, 14), (14, 15), (14, 16), (13, 16), (13, 17)]

Coste: 78

Generados: [(5, 1), (4, 2), (5, 2), (6, 2), (4, 3), (5, 3), (6, 3), (4, 4), (5, 4), (6, 4), (7, 2), (7, 3), (7, 4), (8, 2), (8, 3), (8, 4), (3, 2), (3, 3), (3, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (2, 2), (2, 3), (2, 4), (11, 2), (11, 3), (11, 4), (11, 5), (10, 6), (11, 6), (10, 7), (11, 7), (10, 8), (11, 8), (10, 9), (11, 9), (9, 10), (10, 10), (11, 10), (9, 11), (10, 11), (11, 11), (9, 12), (10, 12), (11, 12), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (12, 9), (12, 10), (12, 11), (12, 12), (13, 2), (13, 3), (13, 4), (8, 10), (8, 11), (13, 10), (13, 11), (13, 12), (8, 12), (14, 2), (14, 3), (14, 4), (14, 9), (14, 10), (14, 11), (14, 12), (14, 13), (7, 10), (7, 11), (7, 12), (15, 2), (15, 3), (15, 4), (15, 9), (15, 10), (15, 11), (15, 12), (15, 13), (13, 14), (14, 14), (15, 14), (13, 15), (14, 15), (15, 15), (13, 16), (14, 16), (15, 16), (13, 17), (12, 14), (12, 15), (12, 16)]

Analizados: [(5, 1), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (4, 2), (4, 3), (4, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (3, 2), (3, 3), (10, 2), (10, 3), (10, 4), (10, 5), (3, 4), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (11, 2), (11, 3), (11, 4), (11, 5), (11, 6), (11, 7), (11, 8), (11, 9), (11, 10), (11, 11), (11, 12), (2, 2), (2, 3), (2, 4), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (9, 10), (12, 9), (12, 10), (12, 11), (12, 12), (9, 11), (9, 12), (13, 2), (13, 3), (13, 4), (13, 10), (13, 11), (13, 12), (8, 10), (8, 11), (14, 2), (14, 3), (14, 4), (8, 12), (14, 10), (14, 11), (14, 12), (14, 13), (14, 14), (14, 15), (14, 16), (13, 14), (13, 15), (13, 16), (13, 17)]

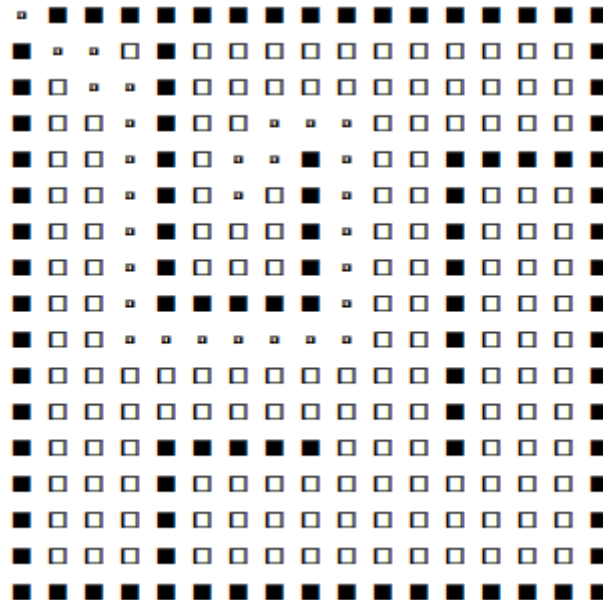


M_2 ID 2 Filas: 17 Columnas: 17 Entrada: (1, 1) Salida: (6, 7) Camino: [(1, 1), (2, 2), (2, 3), (3, 3), (3, 4), (4, 4), (5, 4), (6, 4), (7, 4), (8, 4), (9, 4), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (9, 10), (8, 10), (7, 10), (6, 10), (5, 10), (4, 10), (4, 9), (4, 8), (5, 8), (5, 7), (6, 7)]

Coste: 88

Generados: [(1, 1), (2, 2), (2, 3), (3, 2), (3, 3), (2, 4), (3, 4), (4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (11, 2), (11, 3), (11, 4), (11, 5), (10, 6), (11, 6), (12, 2), (12, 3), (10, 7), (11, 7), (12, 4), (12, 5), (10, 8), (11, 8), (12, 6), (13, 2), (13, 3), (12, 7), (10, 9), (11, 9), (13, 4), (12, 8), (9, 10), (10, 10), (11, 10), (14, 2), (14, 3), (14, 4), (12, 9), (9, 11), (10, 11), (11, 11), (8, 10), (8, 11), (12, 10), (15, 2), (15, 3), (7, 10), (7, 11), (15, 4), (6, 10), (6, 11), (9, 12), (10, 12), (11, 12), (12, 11), (8, 12), (5, 10), (5, 11), (16, 2), (16, 3), (13, 10), (7, 12), (16, 4), (6, 12), (4, 9), (4, 10), (4, 11), (12, 12), (13, 11), (5, 12), (3, 9), (3, 10), (3, 11), (3, 8), (4, 8), (5, 8), (4, 12), (13, 12), (3, 7), (4, 7), (5, 7), (6, 7), (6, 8), (14, 9), (14, 10), (14, 11), (3, 6), (4, 6), (5, 6), (6, 6)]

Analizados: [(1, 1), (2, 2), (2, 3), (3, 2), (3, 3), (2, 4), (3, 4), (4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (7, 2), (7, 3), (7, 4), (8, 2), (8, 3), (8, 4), (9, 2), (9, 3), (9, 4), (10, 2), (10, 3), (10, 4), (10, 5), (11, 2), (10, 6), (11, 3), (11, 4), (10, 7), (11, 5), (12, 2), (11, 6), (10, 8), (12, 3), (12, 4), (11, 7), (12, 5), (10, 9), (13, 2), (13, 3), (12, 6), (11, 8), (13, 4), (12, 7), (10, 10), (9, 10), (11, 9), (14, 2), (8, 10), (14, 3), (12, 8), (7, 10), (14, 4), (10, 11), (11, 10), (9, 11), (6, 10), (15, 2), (12, 9), (8, 11), (15, 3), (7, 11), (5, 10), (11, 11), (10, 12), (15, 4), (12, 10), (9, 12), (6, 11), (16, 2), (8, 12), (4, 10), (16, 3), (4, 9), (11, 12), (7, 12), (5, 11), (12, 11), (4, 8), (5, 8), (16, 4), (13, 10), (6, 12), (4, 7), (5, 7), (6, 7)]



M_3 ID 1 Filas: 5 Columnas: 5 Entrada: (5, 5) Salida: (2, 1) Camino: [(5, 5), (4, 5), (3, 5), (2, 5), (2, 4), (1, 4), (1, 3), (1, 2), (2, 2), (2, 1)]

Coste: 27

Generados: [(5, 5), (4, 4), (4, 5), (5, 4), (4, 3), (3, 5), (2, 4), (2, 5), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (2, 1), (3, 1)]

Analizados: [(5, 5), (5, 4), (4, 5), (4, 4), (4, 3), (3, 5), (2, 5), (2, 4), (1, 5), (1, 4), (1, 3), (1, 2), (2, 2), (2, 1)]



M_3 ID 2 Filas: 5 Columnas: 5 Entrada: (5, 3) Salida: (5, 1) Camino: [(5, 3), (4, 3), (4, 4), (4, 5), (3, 5), (2, 5), (2, 4), (1, 4), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]

Coste: 43

Generados: [(5, 3), (4, 3), (4, 4), (5, 4), (4, 5), (3, 5), (2, 4), (2, 5), (1, 4), (1, 5), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]

Analizados: [(5, 3), (4, 3), (5, 4), (4, 4), (4, 5), (3, 5), (2, 5), (2, 4), (1, 5), (1, 4), (1, 3), (1, 2), (2, 2), (3, 1), (4, 1), (5, 1)]



2.3. Conclusión

En general, la función heurística euclidiana (basada en la recta real desde el punto A hasta el punto B) es menos efectiva que la distancia de Manhattan (genera más nodos) consiguiendo ambas la solución óptima teniendo el coste diagonal igual a 7 y la horizontal/vertical igual a 3.