

Proyecto 1

Estudiantes:

Moises Higuerey Hernandez
Valeria Quesada Benavides
Josué Alvarado Mares
Antony Artavia Palma

Profesor:

Jorge Vargas Calvo

A*

Inteligencia Artificial (IC-6200)

I Semestre, 2022

Instituto Tecnológico de Costa Rica

Definición del estado del espacio	2
Unidad de medida del costo de transición	2
Función de costo de transición de estado	2
Función de costo estimado futuro para cada estado	2
Demostración de los requisitos 1 y 2 del teorema de admisibilidad	3
Justificación del optimismo de la función de costo estimado futuro	3
Tiempo que tarda el programa en resolver los problemas según la dificultad	4
Fácil	4
Medio	6
Difícil	8

Definición del estado del espacio

Un estado en el espacio está constituido por la posición en el tablero del *coche rojo*¹ y los demás coches en el momento, por lo que cada movimiento hecho por un vehículo crearía un nuevo estado, pues sería una nueva configuración en las posiciones de los vehículos en el tablero.

Unidad de medida del costo de transición

La unidad de medida escogida es la cantidad de movimientos acumulados de los vehículos entre un estado y otro.

Función de costo de transición de estado

En nuestro caso tenemos $c(N_i, N_j) = 1$, pues se tomó la decisión que por simplicidad los vehículos se desplazarán un máximo de un espacio a la vez, independientemente de la posibilidad de un vehículo de poder desplazarse más de un espacio, considerando que aún con esta decisión, ese vehículo (con tal posibilidad) en los estados abiertos eventualmente recorrerá todas esas posibilidades de movimiento.

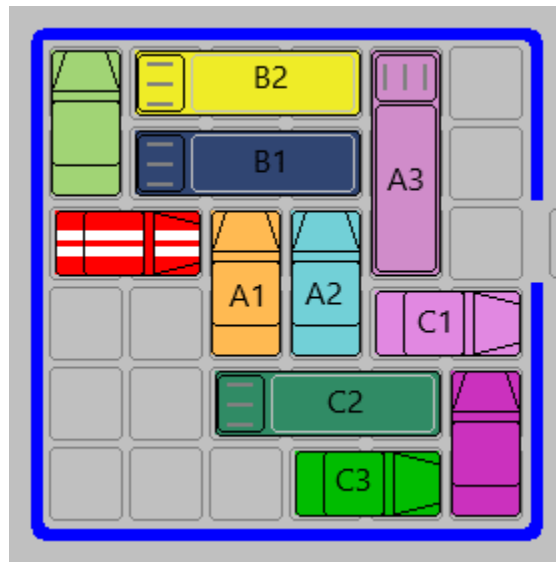
Función de costo estimado futuro para cada estado

La función de costo estimado futuro $F(n)$ está constituida por el acumulado de movimientos desde el estado inicial $G(n)$ y la heurística $H(n)$ es un cálculo de la suma entre la cantidad de vehículos bloqueando al carro rojo para salir y el mínimo de vehículos bloqueando a estos vehículos (por arriba, abajo, izquierda o derecha).

A continuación, un ejemplo donde se explica la función de costo estimado con la imagen de un estado en un tablero.

¹ El coche rojo es el coche que tiene que salir para completar el juego.

Ejemplo con ilustración.



Se cuentan los vehículos que bloquean al coche rojo, en este caso son 3 (A1, A2 y A3) y se procede a contar el mínimo de carros que bloquean a estos (arriba o abajo en este caso):

- En caso de A1, lo bloquean 2 por arriba (B1 y B2) y 1 por debajo (C2). El mínimo sería 1.
- En caso de A2, lo bloquean 2 por arriba (B1 y B2) y 2 por debajo (C2 y C3). El mínimo sería 2.
- En caso de A3, lo bloquean 0 por arriba y 3 por debajo (C1, C2 y C3). El mínimo sería 0.

Lo que daría:

$$G(n) = 3$$

$$H(n) = 1 + 2 + 0 = 3$$

Y quedaría finalmente:

$$F(n) = 3 + 3 = 6$$

Demostración de los requisitos 1 y 2 del teorema de admisibilidad

El algoritmo que planteamos cumple con el requisito 1, pues en cada nodo puede ocurrir a lo sumo (suponiendo que ningún vehículo tenga bloqueos) un movimiento por cada vehículo en el tablero, además el costo $c(N_i, N_j)$ de cada arco del grafo siempre va a ser mayor o igual que 1.

Justificación del optimismo de la función de costo estimado futuro

El momento ideal del tablero es cuando el coche rojo tiene todo el camino a la salida despejado (cero vehículos bloqueando su camino), así realmente se puede determinar informalmente que la función es optimista, pues cuando muevo un vehículo este podría bloquear al coche rojo o bloquear a otros a otros vehículos que necesito mover para despejar el camino del coche rojo, por lo tanto, en lo que muevo vehículos puedo bloquear el camino de otros que ocupan moverse y alejarme de la solución, lo que aumenta la cantidad de movimientos que necesito para llegar a la solución, por lo que en la práctica se requerirían realizar mínimo la cantidad de movimientos determinados por \hat{h} .

Tiempo que tarda el programa en resolver los problemas según la dificultad

Con los problemas que se realizaron las pruebas se obtuvieron los siguientes resultados. Se muestran mediante imágenes y en ellas se encuentra una columna de texto al lado derecho. Las dos primeras líneas hacen referencia a los movimientos mientras que la dos últimas en el tiempo que el programa utilizó para realizar el cálculo.

Fácil

Se utilizaron niveles brindados por el profesor y uno extraído del juego original.

Caso #1

Rush Hour

			6	6	
2	2	1			
3		1	7	7	
3	5	5		8	
3				8	
4	4			8	

Problem solved in:
19 movements
Seconds it took to solve it:
1.1982505321502686

Start

Caso #2

Rush Hour

	3				
	3	5	5	5	
	3		1	7	7
	4	4	1		
				8	9
2	2	6	6	8	9



Problem solved in:

7 movements

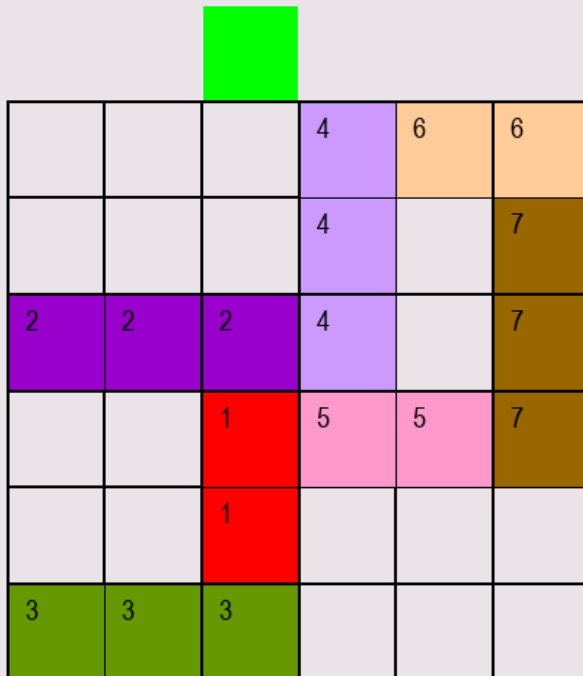
Seconds it took to solve it:

0.016999244689941406

Start

Caso #7

Rush Hour



A 6x6 grid representing a Rush Hour puzzle. The grid contains cars of various colors and sizes, each with a number indicating its length. A green square is positioned above the top row, centered over the third column. The cars are as follows:

			4	6	6
			4		7
2	2	2	4		7
		1	5	5	7
		1			
3	3	3			

Problem solved in:

23 movements

Seconds it took to solve it:

0.39092206954956055

Start

Medio

Se utilizaron niveles brindados por el profesor y uno extraído del juego original.

Caso #3

Rush Hour

	3				8
	3	5	5	5	8
1	1	6			9
		6			9
2	4	4			9
2				7	7

Problem solved in:
10 movements

Seconds it took to solve it:
0.007997751235961914

Start

Caso #4

Rush Hour

2	4	5	7	9	9
2	4	5	7	10	10
			1	1	12
3	3			11	12
		6	8	11	13
		6	8	11	13

Problem solved in:

38 movements

Seconds it took to solve it:

73.81212568283081

Start

Caso #8

Rush Hour

	4	4	9	9	
2	2	7	10	12	14
3	5	7	10	12	14
3	5	1	1	12	14
3	5	8	8	13	13
	6	6	11	11	

Problem solved in:

33 movements

Seconds it took to solve it:

0.9185824394226074

Start

Difícil

Se utilizaron niveles brindados por el profesor y uno extraído del juego original.

Caso #5

Rush Hour

	4	5	5	9	9
	4		7	10	11
1	1		7	10	11
2	2			10	
3		6	8	8	12
3		6			12

Problem solved in:
27 movements

Seconds it took to solve it:
3.090674877166748

Start

Caso #6

Rush Hour

2	2		7	9	9	
3			7	10	10	
3	1	1	7		11	
4		5	8	8	11	
4		5			12	
		6	6	6	12	

Problem solved in:

42 movements

Seconds it took to solve it:

5.447527647018433

Start

Caso #12

Rush Hour

2			1	3	3
2	4	4	1	5	6
7	7	8		5	6
9	9	8		10	
	11	12	12	10	13
	11	14	14	14	13

Problem solved in:

51 movements

Seconds it took to solve it:

393.36944580078125

Start

Muy difícil

Se utilizaron niveles extraídos del juego original.

Caso #9

Rush Hour

	2	3	3	4	4	
	2	5	5	6		
	2	7		6	8	8
		7	9	1	1	10
		7	9		11	10
	12	12	13	13	11	10

Problem solved in:
58 movements

Seconds it took to solve it:
37.65803003311157

Start

Caso #10

Rush Hour

	2	3	3	3	4	
	2			5	4	
6	1	1		5	7	
6		8	9	9	7	
6		8	10	11	11	
12	12	12	10			

Problem solved in:

57 movements

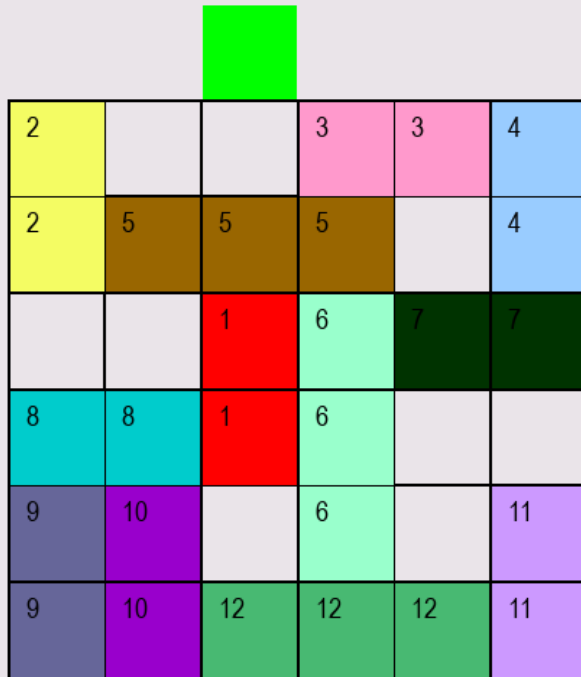
Seconds it took to solve it:

40.42049956321716

Start

Caso #11

Rush Hour



A 6x6 grid representing a Rush Hour puzzle. The grid contains cars of various colors and sizes, each with a number indicating its length. A red car is in the top row, column 3. A yellow car is in the top row, column 1. A pink car is in the top row, column 4. A blue car is in the top row, column 6. A brown car is in the second row, column 2. A light blue car is in the second row, column 6. A red car is in the third row, column 3. A green car is in the third row, column 4. A dark green car is in the third row, column 5. A cyan car is in the fourth row, column 1. A cyan car is in the fourth row, column 2. A red car is in the fourth row, column 3. A green car is in the fourth row, column 4. A grey car is in the fourth row, column 5. A grey car is in the fourth row, column 6. A purple car is in the fifth row, column 1. A purple car is in the fifth row, column 2. A green car is in the fifth row, column 4. A purple car is in the fifth row, column 6. A purple car is in the sixth row, column 1. A purple car is in the sixth row, column 2. A green car is in the sixth row, column 3. A green car is in the sixth row, column 4. A green car is in the sixth row, column 5. A purple car is in the sixth row, column 6.

2			3	3	4
2	5	5	5		4
		1	6	7	7
8	8	1	6		
9	10		6		11
9	10	12	12	12	11

Problem solved in:

65 movements

Seconds it took to solve it:

40.95346736907959

Start